REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
MAY 1995

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
CSDL-T-1242 Estimation of the local Vertical State for A Guided Munition Shell with An Embedded GPS/Micro-Mechanical Inertial NAVIGATION System

5. FUNDING NUMBERS

6. AUTHOR(S)
David J. Lucia

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFIT Students Attending:
Massachussets Institute of Technology

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/CI/CIA
95-015

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DEPARTNEMT OF THE AIR FORCE
AFIT/CI
2950 P STREET, BDLG 125
WRIGHT-PATTERSON AFB OH 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release IAW AFR 190-1
Distribution Unlimited
BRIAN D. GAUTHIER, MSgt, USAF
Chief Administration

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

14. SUBJECT TERMS

15. NUMBER OF PAGES
127

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

CSDL-T-1242

ESTIMATION OF THE LOCAL VERTICAL STATE FOR A GUIDED MUNITION SHELL WITH AN EMBEDDED GPS/MICRO-MECHANICAL INERTIAL NAVIGATION SYSTEM

by

David J. Lucia

May 1995

Master of Science Thesis
Massachusetts Institute of Technology

DRAPER
LABORATORY

The Charles Stark Draper Laboratory, Inc.
555 Technology Square, Cambridge, Massachusetts 02139-3563

Estimation of the Local Vertical State for a Guided Munition Shell with an Embedded GPS/Micro-Mechanical Inertial Navigation System

by

David J. Lucia

B.S.A.S.E, University of Texas
(1988)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

Signature of Author David J Lucia

Department of Aeronautics and Astronautics
May 1995

Approved by ______________________

Tom P. Thorvaldsen
Principal Member Technical Staff, Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by ______________________

Professor John J. Deyst
Thesis Supervisor

Certified by ______________________

Professor Harold Y. Wachman
Chairman, Department Graduate Committee

Estimation of the Local Vertical State for a Guided Munition Shell with an Embedded GPS/Micro-Mechanical Inertial Navigation System

by

David J. Lucia

Submitted to the Department of Aeronautics and Astronautics
on May 12, 1995 in partial fulfillment of the requirements for
the Degree of Master of Science in Aeronautics and
Astronautics

ABSTRACT

This project considers a ballistic munition shell with an automated flight control system commanding control surface deflections to canards installed on the shell nose to guide the munition shell to its target. An algorithm to estimate the local vertical state of the shell spinning between 0 and 2 Hz is developed using GPS velocity measurements and 3 axis micro rate gyro measurements. Roll angle estimation performance is simulated and analyzed. Several shell roll rates between 0 and 2Hz are tested to quantify estimator performance across the specified roll rate frequency spectrum. A priori gravity turn information is substituted for GPS measurements to make an inertial only algorithm and the performance effects are quantified. Finally, an extended Kalman filter implementation of the inertial only roll estimating algorithm is developed and its performance is simulated and analyzed by first isolating the individual contributions of each sensor error and then considering the full suite of errors.

Thesis Supervisor: Dr. John J. Deyst

Title: Professor, Massachusetts Institute of Technology

Technical Supervisor: Tom P. Thorvaldsen

Title: Principal Member Technical Staff, Charles Stark Draper Laboratory, Inc.

Acknowledgments

It is important for me to express credit and appreciation to those who have contributed to the preparation and completion of this thesis. I could never have done this research without their support. First, I must thank Tom Thorvaldsen, Don Gustafson and John Deyst for their technical guidance to this new comer to the world of navigation filtering. Their years of experience guided my enthusiasm to learn which resulted in the research contained herein. I must also thank Ken Spratlin and Doug Fuhry for developing such a nice Monte Carlo simulation for me to modify and use to test various algorithms. I would also like to thank my fellow students Greg Bierman and Chris Stoll for their camaraderie throughout the thesis endeavor. There is no better way to fine tune ideas than to try them on peers. Thanks for listening and helping.

The year of research spent on this thesis happened to coincide with a very difficult period in my personal life and without the help of a few good personal friends, I would not have been able to be affective. Special thanks to Matt and Tamara Goda, Mark and JoAnn Edwards, Jeff and Jody Keillor, David Johnson and my family for being there when I needed them.

Finally, I would be remiss if I did not thank my God, Jesus Christ, for the opportunity to learn, the mind to understand, and the heart to persevere in this endeavor. I would not have anything to study without the powerful hand of the Creator, nor would I be able to prosper and enjoy my work apart form the love of God who has shown that He cares for me personally and deeply.

This thesis was prepared at the Charles Stark Draper Laboratory under Corporate Sponsored Research CSR-319.

Publication of this report does not constitute approval by The C.S. Draper Laboratory of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

David J. Lucia

Table of Contents

Chapter **Page**

List of Figures

Figure **Page**

List of Tables

Chapter 1

Introduction

The improvements over the past decade in micro inertial sensor technology at the Charles Stark Draper Laboratory have made feasible the development of micro inertial navigation systems. Vehicles whose small size prohibited installation of the latest sensor technology available for larger platforms, are now ripe for navigation and control enhancement. The ballistic artillery shell is such a platform and is the subject of this thesis.

This research is a precursor to the competent munitions program at Draper Laboratory directed by the Naval Surface Fire Support Programs Office and called the Guided Munitions Demonstrations Program. The project involves development of a Guidance and Navigation package to be integrated into the cylindrical body of a 5 inch diameter naval gun shell. The adapted shell is fitted with canards as control surfaces and is called the Guided Munition Round (GMR, see Figure 1.1).

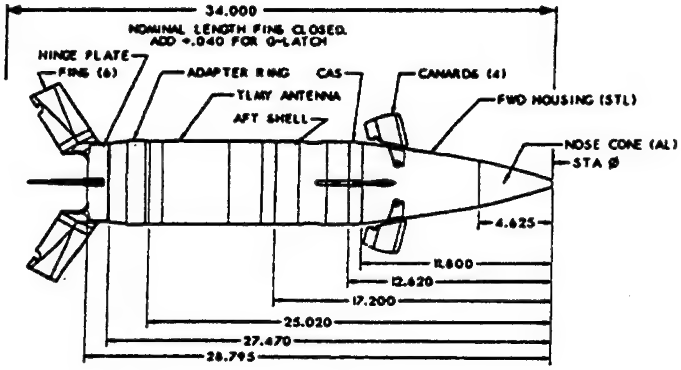


FIGURE 1.1 Guided Munition Round

Draper is under contract to build five Guidance and Navigation boards and deliver them to the Navy for firing at a test range. Once the risk reducing Guided Munition Demonstration Program is complete, a follow-on program is planned to integrate the micro sensors and GPS based guidance system into the smaller nose cone section of the shell [1] .

1.1 Guided Munition Round Requirements

To this date, artillery shell targeting systems delivered unguided munitions to target by using a priori knowledge of shell barrel exit velocity, shell aerodynamics, and wind estimates to aim the gun barrel in azimuth and elevation at a known target location. Years of perfecting these targeting techniques yield a Circular Error Probable (CEP) of about 330 feet for a 7.5 mile shot at a stationary target. This means that 50% of the shots fired will land within 330 feet of the target at a range of 7.5 miles. The accuracy goal for the GMR is 50 feet CEP for the same 7.5 mile shot. In addition to CEP performance of the overall GN&C system, there is a technology demonstration requirement for the electronics to simply survive the launch environment of 6200g (lasting approximately 10ms) and operate throughout the trajectory. [1]

1.2 Guided Munition Round Time Line

The Guided Munition system concept is based upon a series of events that follow a specific time line shown in Figure 1.2 . The GPS receiver will be allowed to acquire all satellites available on the ground prior to launch. Fast reacquisition of the GPS P-code accuracy signals will occur during the flight. P-code accuracy is assumed because this is a military application.

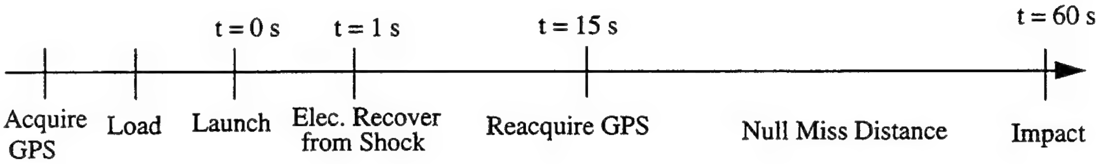


FIGURE 1.2 GMR Flight Time Line

The GMR exits the barrel of the gun 25ms after launch and spinning at approximately 25Hz. The fins and canards deploy and latch within 30m of the barrel (or about 0.01sec). The shell subsequently de-spins to within 0-2Hz in about 1 second. The GPS/INS is inactive for the first second of the flight and then reacquisition of the GPS signal is begun along with the local vertical estimation routine which is the subject of this thesis. GPS reacquisition is complete in about 15 seconds and the Guidance, Navigation and Control (GN&C) system makes course corrections for the rest of the trajectory (about 45 seconds). [1]

1.3 Micro Inertial Navigation

A Micro Inertial Measurement Unit (MIMU) consists of 3 each orthogonally mounted micro gyros and micro accelerometers. Cost per unit of a MIMU depends on the accuracy required of the gyros and accelerometers. If a navigation system design can tolerate reasonable system errors, then gyro and accelerometer manufacturing is relaxed resulting in relatively inexpensive MIMUs. However, if the navigation system requires greater accuracies from the noisy micro sensors the cost per unit begins to climb.

A typical aircraft inertial navigation system consisting of 3 each orthogonally mounted ring laser gyros and accelerometers is far too large for integration into the guided munition shell platform. The most accurate gyros and accelerometers are not only too large for the artillery shell application, but the cost per unit of such systems is too high when considering the "use once and destroy" mission scenario of the artillery shell autopilot. Finally, the larger rate gyros are not sturdy enough to withstand the high g-loads experienced by the artillery shell platform.

The MIMU accelerometer cube is 2" x 2" x 2" for the Guided Munitions Demonstration Program. The hybrid micro gyros require 3 boards 4" x 4" x 0.75" each. Total volume for the MIMU is 44 cubic inches (or roughly a 6.6" cube). The MIMU is made of relatively light electronic components so the total weight is on the order of ounces. Electric power to operate the MIMU on the order of 60 watts and the MIMU target cost is around $5000. [1]

The MIMU micro gyro performance requirements are described in terms of scale factor error, g sensitivity, misalignment, non-orthogonality, and angle random walk. The micro accelerometer performance requirements are described in terms of scale factor error, g^2 sensitivity, misalignment, non-orthogonality, and velocity random walk. These are described in detail in Chapter 6 on page 98.

1.4 GPS Navigation

Embedding a GPS receiver in the artillery shell will ease the accuracy requirements on the MIMU. The ability to receive GPS signals on board an artillery shell and transpond to a ground station has been demonstrated in Army tests and documented in *GPS World* and other papers [11] , [2] . Four patch antennas will be mounted at equidistant locations around the shell's cylindrical body. A multi-channel GPS receiver is embedded to provide position and velocity (initial program) or psuedorange and delta range (follow-on program) to the navigation system throughout the flight.

1.5 The Initialization Problem

The g-load on the sensors at launch will pin the micro sensors against their maximum deflection points and effectively corrupt any state information from these sensors. Once out of the barrel, the MIMU and GPS receiver should operate as specified, but the initial state information will be lost. Before a navigation system can begin to feed state information into a control algorithm to guide the shell, an estimate of the initial state starting at some time after leaving the barrel must be made to initialize the transformation from body to local level coordinate frames. This initial state estimate must be provided with enough time remaining in the flight for the initialized navigation routine and control system to correct the shell trajectory.

The a priori knowledge used to target the shell in the absence of GN&C becomes useful in this state initialization problem [4] . The azimuth and elevation of the barrel along with a crude estimate of a gravity turning ballistic trajectory can provide decent initial state estimates of the shell pitch and yaw angles. These angles, knowledge of the shell exit velocity (known to a high degree of accuracy from years of Army and Navy testing), and the location of the gun barrel can yield reasonable initial estimates of shell position and velocity. However, determining the shell roll angle defies such an easy solution and is required to find the local vertical state. The local vertical state is defined as a body centered coordinate frame with the “z” axis always pointing down toward the center of the earth. Without knowledge of the shell attitude within the local vertical state, any control surface deflections commanded to the canards attached to the surface of the spinning GMR would result in erroneous course corrections.

However, ample information is available from the installed sensors to estimate the shell roll angle to the required degree of accuracy. It is estimated that an initial error of 30 degrees reducing to about 6 degrees of error at the end of the flight will allow the GN&C to meet the 50 ft. CEP performance goal [4] . The research presented here focuses on development of a local vertical estimation algorithm and analysis of its performance in the presence of various error sources.

Chapter 2

GPS Velocity Measurements to Estimate the Pitch and Yaw Rates of a Non-Spinning Shell

Before proceeding with the shell roll angle estimation problem, an acceptable estimate of the shells pitch and yaw rates needs to be produced. This is foundational to the latter sections because it will be shown that combining lateral rate gyro data (y and z axis rate gyros in the body fixed frame) with these estimates of shell pitch rate and yaw rate yields a measurement of the shell roll angle. With this motivation in mind, this chapter endeavors to use information from GPS velocity data (which is available once per second) to estimate the shell pitch and yaw rate in real time during the trajectory.

2.1 A Look at the Non-Spinning Trajectory

A ballistic trajectory of a non-spinning artillery shell was modeled for the analysis in this chapter. Two independent software simulations were developed to model the artillery shell navigation environment. The first simulation generates a trajectory file of shell position, velocity, acceleration, specific force, attitude quaternion, angular rate and angular acceleration at a 1000Hz data rate. The second simulation incorporates models of GPS and the MIMU with error sources, and navigation algorithms to simulate navigation system performance. The 1000Hz sensor information rate was chosen for this thesis to demonstrate performance when the shell spin rate is 2Hz. A complete description of this simulation is given in Chapter 6 . The trajectory simulated here has a 0 degree azimuth angle (due north) and a 38.75 degree elevation angle. The initial velocity out of the barrel is 830 m/s. The shell position in North East Down (NED) coordinate frame is given in Figure 2.1 .

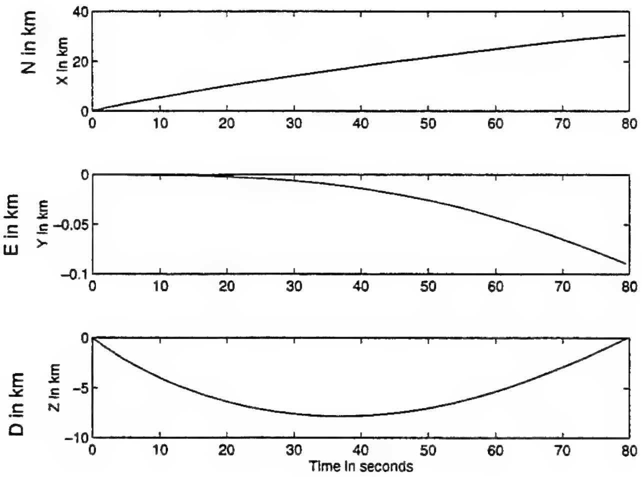


FIGURE 2.1 Non-Spinning Artillery Shell Position in NED

Notice the left turn the shell takes due to the Coriolis effect on the moving shell in a rotating coordinate frame. The non-spinning shell attitude under these conditions is given in Figure 2.2 .

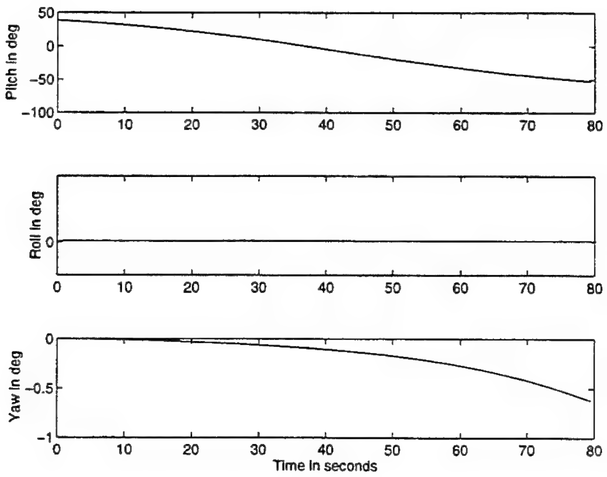


FIGURE 2.2 Artillery Shell Attitude for a Non-Spinning Trajectory

The ballistic trajectory shows the artillery shell pitching down throughout the flight as well as yawing to the left. Since the pitch rate is always acting in the negative "East" direction in NED and the Yaw Rate always acts in the negative "Down" direction in NED, these rates both contain information about the local level direction. The Pitch and Yaw rates for the non-spinning trajectory are given in Figure 2.3 .

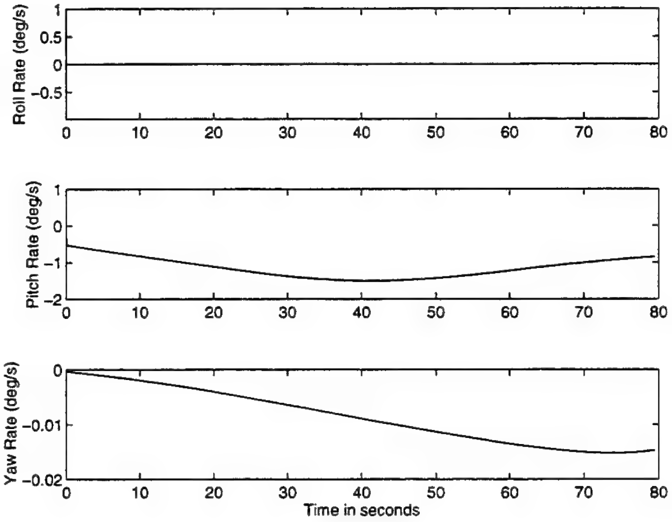


FIGURE 2.3 Non-Spinning Pitch and Yaw Rates

This information will be exploited by the initialization algorithm to estimate the shell roll angle when the shell is spinning.

2.2 Velocity Estimation at 1000Hz from GPS Velocity Measurements at 1Hz

As will be shown later, the GPS velocity data contains information about the non-spinning pitch and yaw attitudes; however, the GPS update rate is only 1Hz. The velocity profile of the shell shown in Figure 2.4 shows the changes in velocity due to aerodynamic drag and gravity. Early in the trajectory, a one second time interval can see as much as 50 m/s velocity change. Clearly, a one second time step is too large to assume velocity is merely constant. Therefore, linear least squares data interpolation is used to provide the velocity estimate across the time step.

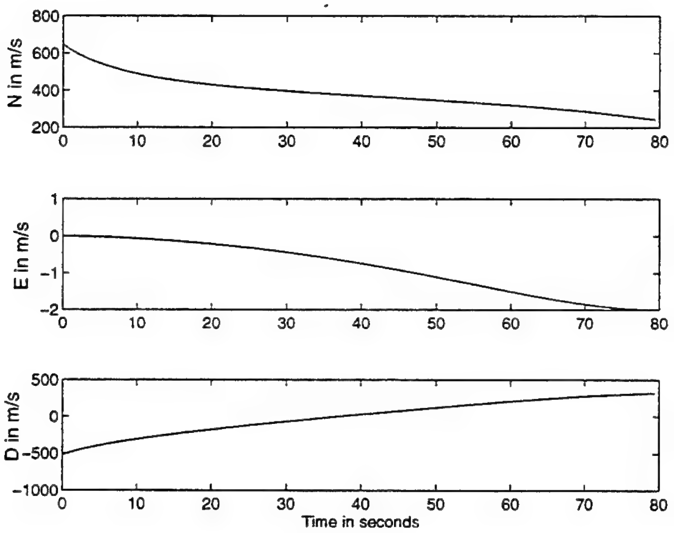


FIGURE 2.4 Shell Velocity Profile in NED

Third order linear least squares curve fitting was implemented into the software simulation to provide a 3rd order polynomial estimate of the velocity profile between the one second GPS velocity updates. Nth order least squares curve fitting is a technique to find the Nth order polynomial that best fits a set (or "batch") of data collected over a certain period of time. A complete description of this technique can be found in reference [8] . Six consecutive GPS velocity measurements were collected sequentially to create the least squares 3rd order polynomial best fitting the 6 GPS data points. After the initial 6 seconds to fill the batch processing register, each additional second included a new GPS velocity measurement, eliminated the oldest GPS velocity measurement in the register, and recalculated the 3rd order polynomial coefficients using the latest 6 data points. During the 1 second intervals between GPS measurements, velocity is estimated using time and the 3rd order least squares polynomial coefficients. The results of this method in the absence of sensor errors is given below in Figure 2.5 .

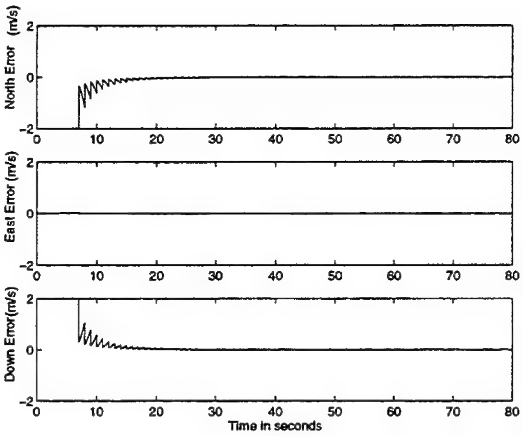


FIGURE 2.5 Velocity Error Using 3rd Order Least Squares Curve Fit (No other errors)

The “saw tooth” effect is due to the error between truth and the 3rd order polynomial curve fit provided by least squares estimation. The errors level out in time revealing that the 3rd order polynomial curve fit is more accurate as the velocity profile smooths out with time along the 80 second trajectory. However, GPS velocity measurements are not without errors. Noise was added ($\sigma = 0.1$ m/s) to the GPS data and the resulting error is shown in Figure 2.6 .

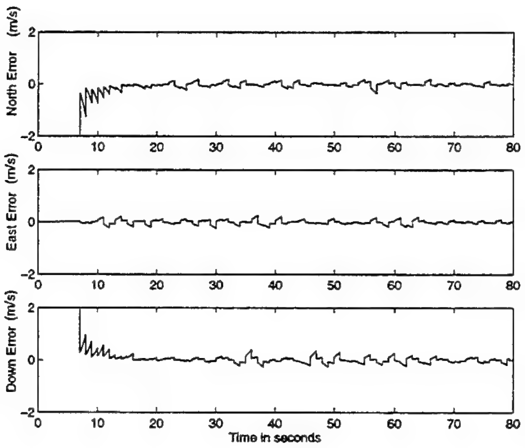


FIGURE 2.6 Velocity Error Using 3rd Order Least Squares Curve Fit and GPS Noise at $\sigma = 0.1$ m/s

2.3 GPS Euler Angle Projection

Now that a reliable velocity estimate is available at the system rate of 1000Hz, this velocity measurement can be used to determine the non-spinning shell pitch and yaw attitude. First, an assumption is made to simplify the analysis. It is assumed that the roll (or "x") axis in the body frame is aligned with the velocity vector. In reality, artillery shells usually achieve some angle of attack. Implementation of the following technique will require additional complexity to adjust for angle of attack. With this assumption made, the velocity vector components in the NED frame are related to the shell attitude as illustrated in Figure 2.7 .

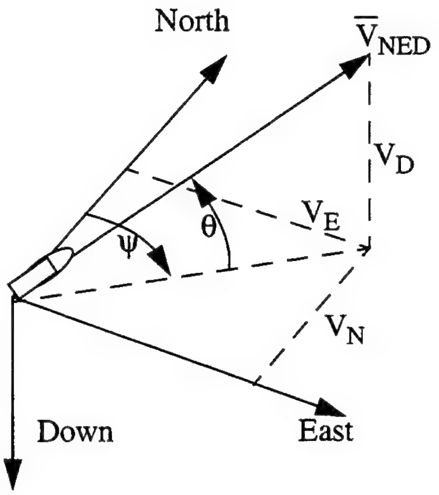


FIGURE 2.7 GPS Euler Angle Projection in North East Down Coordinates

The equations relating velocity to yaw and pitch angle are as follows. The terms ψ, and θ are "hatted" to denote that they are estimates.

$$\tan\hat{\psi} = \frac{V_E}{V_N} \qquad \text{EQ (2.3.1)}$$

$$\tan\hat{\theta} = \frac{V_D}{\sqrt[2]{V_N^{\ 2} + V_E^{\ 2}}} \qquad \text{EQ (2.3.2)}$$

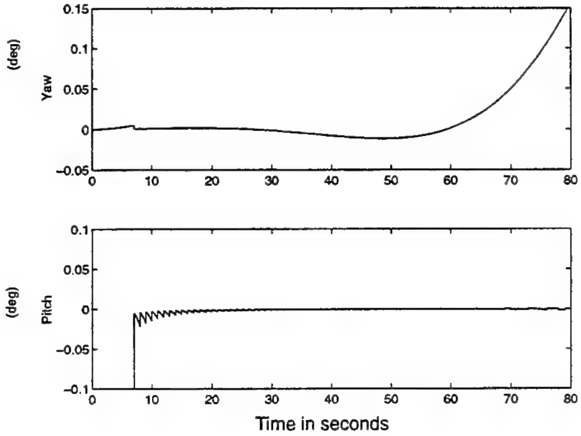


FIGURE 2.8 GPS Euler Angle Projectile Error

In the absence of errors, the error in estimated Euler angles using this technique are presented above in Figure 2.8 . The results with GPS velocity noise set at σ = 0.1 m/s is given in Figure 2.9 .

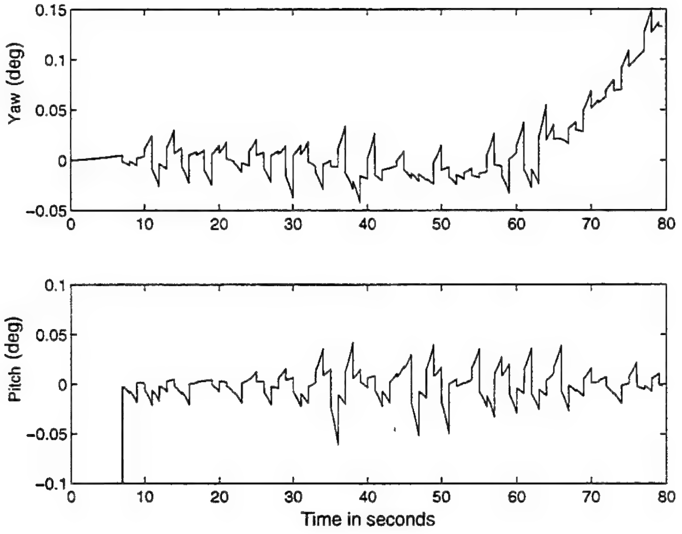


FIGURE 2.9 GPS Euler Angle Projection Error with Velocity Noise σ = 0.1m/s

2.4 Estimating Pitch and Yaw Rates of the Non-Spinning Shell

Pitch and yaw rates are obtained by dividing the estimated pitch and yaw angles by the time step determined by the data processing rate. The equations for estimated pitch and yaw rate (Q and R respectively) are as follows.

$$\hat{Q} = \frac{\Delta\hat{\theta}}{\Delta t} \qquad \text{EQ (2.4.1)}$$

$$\hat{R} = \frac{\Delta\hat{\psi}}{\Delta t} \qquad \text{EQ (2.4.2)}$$

Estimates of Q and R can be made by differentiating (across the 1000Hz time step) the change in estimated yaw and pitch angles estimated in turn by GPS Euler angel projection. The results of this technique are presented in Figure 2.10 without GPS velocity noise, and in Figure 2.11 with the GPS velocity noise of $\sigma = 0.1$ m/s.

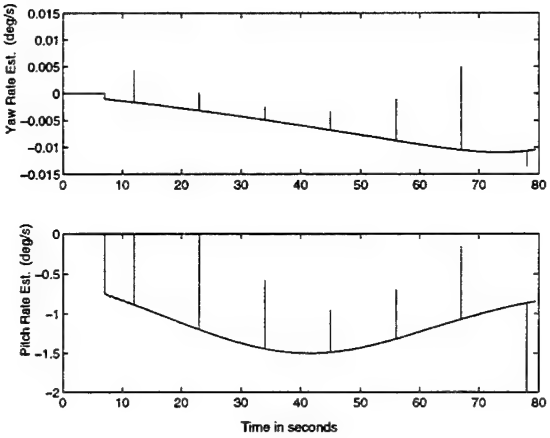


FIGURE 2.10 Non-Spinning Pitch and Yaw Rate Estimates (No Noise)

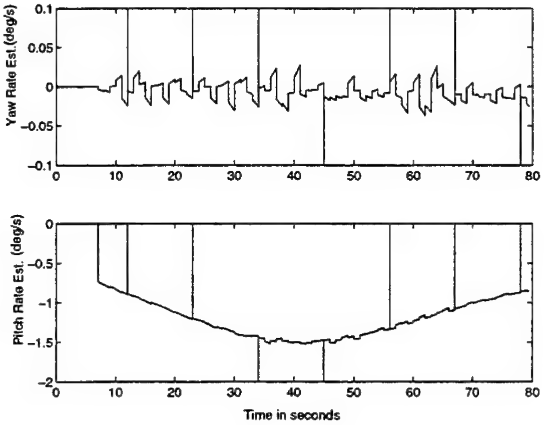


FIGURE 2.11 Non-Spinning Pitch and Yaw Rate Estimates (deg/sec) with GPS Velocity Noise $\sigma = 0.1$ m/s

The spikes that appear periodically in Figure 2.10 and Figure 2.11 are peculiarities due to the software implementation of this routine. They are inconsequential to the performance of the navigation algorithm developed in later sections because these algorithms will be insensitive to the pitch and yaw rate magnitudes provided by GPS Euler Angle Projection.

Chapter 3

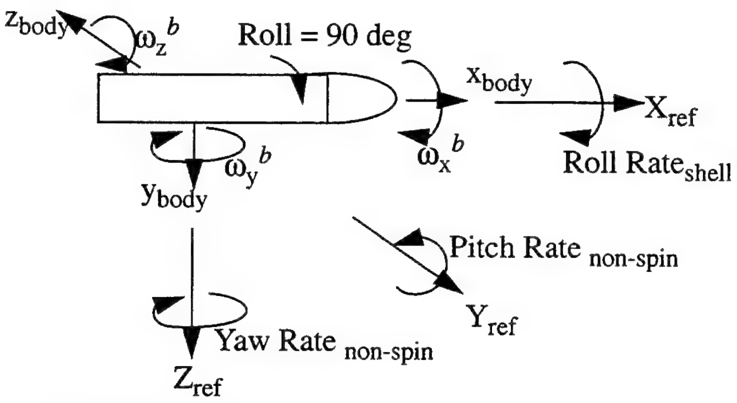
Estimating the Roll Angle of a Spinning Shell

This chapter focuses on estimating the roll angle of a shell spinning at 2Hz. The problem must be solved without prior knowledge of the shells initial roll angle. The chapter begins by showing that the lateral rate gyros (or the y and z axis rate gyros mounted in the body frame of the spinning shell) can be combined to provide measurements of the sine and cosine of the shell roll angle. However, the gyros are very noisy. The next portion of the chapter is devoted to deriving a least squares estimator that can extract the sine and cosine of roll from the noisy measurements. This estimator employs some trigonometric identities to relate the measurements to a linear combination of known and unknown quantities. The known quantity turns out to be the sine and cosine of the roll rate multiplied by time and the unknown quantity is the sine and cosine of the phase of the roll angle. Since the relationship between measurements and unknowns is made linear, a deterministic least squares curve fitting approach is applied to estimate the unknown quantities. With this done, the roll angle can easily be found by applying the arctangent function. The last part of the chapter is intended to analyze the performance of this method in the presence of a variety of rate gyro errors.

To start, it is assumed that the spinning shell follows the same trajectory as the non-spinning shell for shell roll rates ranging from 0Hz to 2Hz. If atmospheric effects other than drag were considered in the trajectory model, a spinning shell encounters an additional yawing moment due to the interaction between the air stream and the spinning boundary layer [5] . The effect is called the yaw of repose and is ignored for this analysis because it is negligible at the small roll rates being considered [3] .

3.1 Roll Information Imbedded in Pitch and Yaw Axis Rate Gyros

As spin about the shell nose is introduced into the trajectory, the body frame rotates in relation to the non-spinning reference frame at the roll rate. The pitch and yaw rate gyros mounted in the spinning body axis will detect the non-spinning pitch and yaw rates once per revolution when the roll angle is identically zero. As the shell rolls through 360 degrees the angular rate measured by each will vary sinusoidally with a period equal to the period of the shell spin. This is illustrated below in Figure 3.1 .



when Roll = 90 deg...

$$\omega_x^b = \text{Roll Rate}_{shell}$$

$$\omega_y^b = \text{Yaw Rate}_{non\text{-}spin}$$

$$\omega_z^b = -\text{Pitch Rate}_{non\text{-}spin}$$

FIGURE 3.1 Angular Rates in the Rotating Body Frame Compared to a Non-Spinning Ref. Frame

The angular rates sensed by the rate gyros mounted in the y and z body axes are given by the following formulas. Here ϕ is the roll angle, and Q and R represent the pitch and yaw rate respectively in the non-spinning reference frame.

$$\omega_y^b = Q\cos\phi + R\sin\phi \qquad \text{EQ (3.1.1)}$$

$$\omega_z^b = R\cos\phi - Q\sin\phi \qquad \text{EQ (3.1.2)}$$

In the absence of errors from the rate gyros, the detected y and z angular rates in the body frame with the spin rate P = 1/4 Hz are shown in Figure 3.2 .

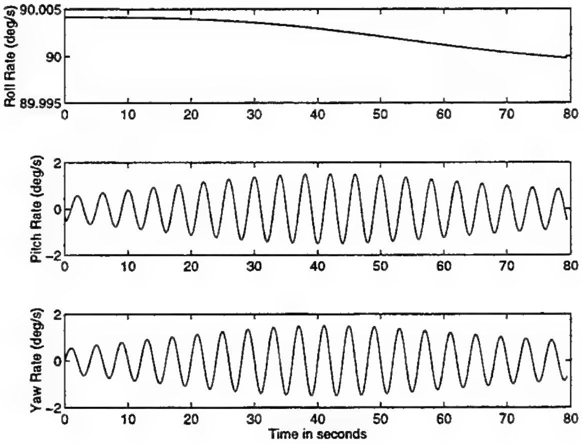


FIGURE 3.2 Spinning Shell Angular Rates in Body Axes

3.2 Roll Estimation Using Linear Least Squares

Solving EQ 3.1.1 and EQ 3.1.2 for sinϕ and cosϕ (where ϕ is time varying) produces the following relationships.

$$\cos\phi(t) = \frac{Q\omega_y^{\,b} + R\omega_z^{\,b}}{Q^2 + R^2} \qquad \text{EQ (3.2.1)}$$

$$\sin\phi(t) = \frac{R\omega_y^{\,b} - Q\omega_z^{\,b}}{Q^2 + R^2} \qquad \text{EQ (3.2.2)}$$

Combining the lateral rate gyro measurements and using the estimates of R and Q, as given in Section 2.4, produces processed measurements of cosϕ(t) and sinϕ(t). Initial attempts at extracting the roll angle information from these equations are documented in the Appendices of this report. The linear least squares algorithm presented here works relatively well and will give insight into the Kalman Filter approach presented later. Focusing our attention on EQ 3.2.2 as the measurement of sinϕ(t), the sensor data combined in this way produces a measurement vector at the system rate of 1000Hz. Figure 3.3 shows the sinϕ(t) measurement for a spinning trajectory of 1/4 Hz with gyro noise as specified in Chapter 6. The sinϕ(t) measurements extend outside the -1 to 1 range of a pure sinusoid because of the noise in the rate gyro outputs which are sampled at the system rate of 1000Hz. The variance of the rate gyro noise is also affected by the "Q" and "R" terms in EQ 3.1.2 . This relationship is derived in Section 5.2.1.1.

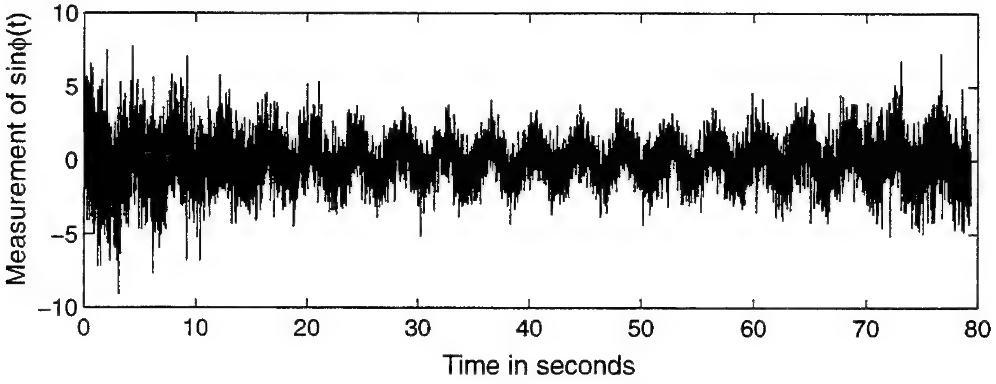


FIGURE 3.3 Sine Algorithm (EQ 3.1.2) Output for P = 1/4 Hz

Sinϕ(t) can be written two ways as shown in EQ 3.2.3 and EQ 3.2.4 where Φ_P is the phase of the sinϕ(t) curve. Notice that EQ 3.2.4 is linear in cosΦ_P and sinΦ_P.

$$y(t) = \sin\left(\omega_x^{\ b} \cdot t + \Phi_P\right) \qquad \text{EQ (3.2.3)}$$

$$y(t) = \sin\left(\omega_x^{\ b} \cdot t\right) \cdot \cos\Phi_P + \cos\left(\omega_x^{\ b} \cdot t\right) \cdot \sin\Phi_P \qquad \text{EQ (3.2.4)}$$

The vector notation for n measurements of y(t)= sinϕ(t) (from EQ 3.1.2) and the matrix form of EQ 3.2.4 at discrete times is provided in EQ 3.2.5 . The equation is written symbolically in EQ 3.2.6 . The roll axis rate gyro provides the Maximum Likelihood (ML) estimate of roll rate "$\omega_x^{\ b}$" which changes with time throughout the trajectory depending on roll axis rate gyro errors. ML estimation of angular rate from noisy gyros is developed in Appendix B .

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} \sin\left(\hat{\omega}_x^{\ b}{}_1 t_1\right) & \cos\left(\hat{\omega}_x^{\ b}{}_1 t_1\right) \\ \sin\left(\hat{\omega}_x^{\ b}{}_2 t_2\right) & \cos\left(\hat{\omega}_x^{\ b}{}_2 t_2\right) \\ \sin\left(\hat{\omega}_x^{\ b}{}_3 t_3\right) & \cos\left(\hat{\omega}_x^{\ b}{}_3 t_3\right) \\ \dots & \dots \\ \dots & \dots \\ \sin\left(\hat{\omega}_x^{\ b}{}_n t_n\right) & \cos\left(\hat{\omega}_x^{\ b}{}_n t_n\right) \end{bmatrix} \cdot \begin{bmatrix} \cos\Phi_P \\ \sin\Phi_P \end{bmatrix} \qquad \text{EQ (3.2.5)}$$

$$\bar{Y}_{nx1} = Z_{nx2} \cdot \bar{\alpha}_{2x1} \qquad \text{EQ (3.2.6)}$$

y_i is the measurement defined in EQ 3.1.2 .

$\hat{\omega}_x{}^b{}_i$ is the ML estimate of roll rate at time t_i from the x axis gyro

Φ_P is the phase as defined in EQ 3.2.3

Deterministic linear least squares curve fitting provides the minimum error estimate of $\bar{\alpha}$ based on the observations $\bar{y}$ and the calculated matrix relationship Z. This estimate is the solution to the normal equations for this system shown in EQ 3.2.7 . [8]

$$\alpha_{LLSE} = \left(Z^T \cdot Z\right)^{-1} \cdot Z^T \cdot Y \qquad \text{EQ (3.2.7)}$$

Once the estimation of the phase parameter for the sinϕ signal is accomplished, we can extract the roll angle using the ML estimate of spin rate ($\omega_x{}^b$) and the following steps.

$$\sin\hat{\phi}(t) = \sin\left(\hat{\omega}_x{}^b \cdot t + \Phi_P\right) \qquad \text{EQ (3.2.8)}$$

$$\cos\hat{\phi}(t) = \cos\left(\hat{\omega}_x{}^b \cdot t + \Phi_P\right) \qquad \text{EQ (3.2.9)}$$

$$\hat{\phi}(t)_{\sin\phi} = \mathrm{atan}\left(\frac{\sin\hat{\phi}(t)}{\cos\hat{\phi}(t)}\right) \qquad \text{EQ (3.2.10)}$$

The entire procedure can be repeated using the cosϕ information from EQ 3.2.1 .

$$\cos\phi(t) = \frac{Q\omega_y{}^b + R\omega_z{}^b}{Q^2 + R^2} \qquad \text{EQ (3.2.11)}$$

The LLSE algorithm is repeated exactly as shown above except the matrix formulation for $\bar{y}$, Z and $\bar{\alpha}$ come from the cosine expansion shown in EQ 3.2.13 .

$$y(t) = \cos\left(\omega_x{}^b \cdot t + \Phi_P\right) \qquad \text{EQ (3.2.12)}$$

$$y(t) = \cos\left(\omega_x{}^b \cdot t\right) \cdot \cos\Phi_P - \sin\left(\omega_x{}^b \cdot t\right) \cdot \sin\Phi_P \qquad \text{EQ (3.2.13)}$$

The average of the two roll angel estimates represents the linear least squares estimate of roll from this algorithm.

$$\hat{\phi}(t)_{LLSE} = \frac{\hat{\phi}(t)_{\sin\phi} + \hat{\phi}(t)_{\cos\phi}}{2} \qquad \text{EQ (3.2.14)}$$

3.2.1 Effect of Micro Gyro and GPS Noise on the LLSE Algorithm

The LLSE algorithm error for a simulated trajectory, gyro noise active, and P = 2 Hz (batch processing time Δt = 10 sec at 1000Hz data rate) is shown in Figure 3.4 .

Clearly, the longer the batch processing time the better the algorithm performance because the filter gets more data over a larger number of cycles to estimate phase. However, not much time is available because the shell trajectory only lasts 80 seconds. The batch processing time of 10 seconds is used throughout this report as a benchmark for analysis and a seemingly reasonable time in considering the entire system

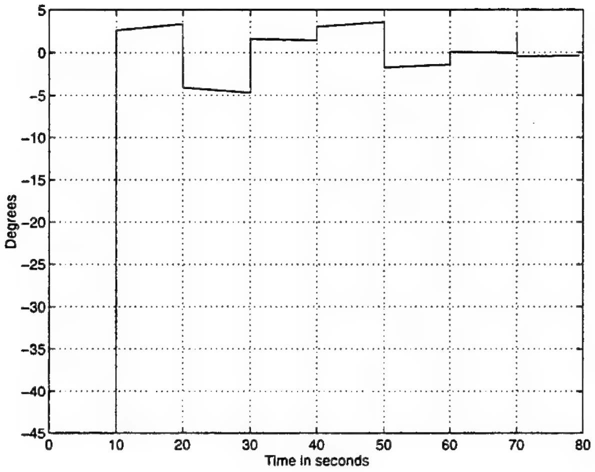


FIGURE 3.4 Roll Error from LLSE Algorithm P= 2 Hz (Data at 1000Hz and Δt = 10 sec)

3.2.2 The Effect of Scale Factor Error on the LLSE Algorithm

Gyro scale factor error produces errant rate indications about all three body axis. Such error in the y and z body axis rate gyros does not affect the algorithm performance because the angular rates are small. Figure 3.5 shows the algorithm performance with gyro noise, GPS noise and y and z axis rate gyro scale factor errors active. The data in the first 17 seconds are aberrations of the software implementation since the algorithm was not initialized until 17 seconds into the simulated trajectory.

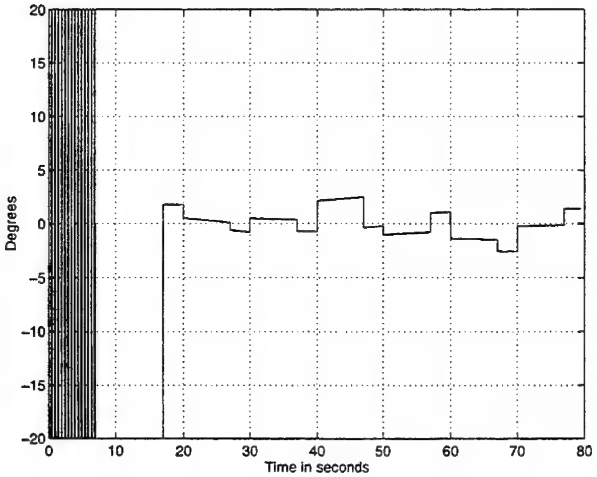


FIGURE 3.5 LLSE Algorithm Error with Y and Z axis Scale Factor Active

Scale factor error in the x body axis rate gyro has a significant effect on algorithm performance. An attempt to estimate and remove x axis scale factor error will be addressed in a later section. Figure 3.6 shows the algorithm performance with gyro noise, GPS noise and x axis rate gyro scale factor error active.

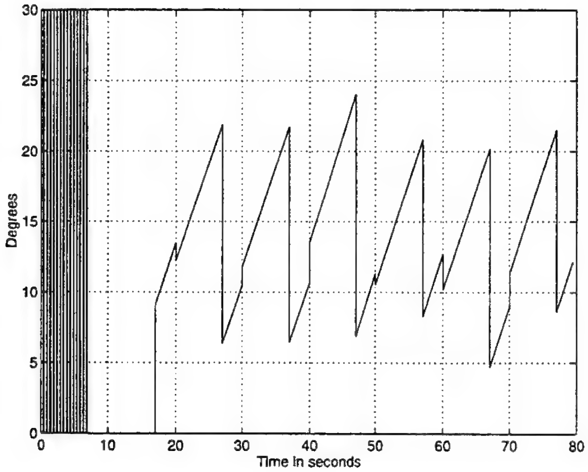


FIGURE 3.6 LLSE Algorithm Error with X Axis Scale Factor Active

Notice that the average error is about 14 degrees. The 2000ppm x axis gyro scale factor error modelled in the simulation produces a 1.44 deg/sec roll rate error (0.004Hz) when the spin rate P = 2Hz. The average error is not surprisingly...

$$xsf \cdot P \cdot \Delta t = \frac{2000ppm}{1000000} \cdot 720\left(\frac{deg}{s}\right) \cdot 10s = 14.4deg \qquad \text{EQ (3.2.1)}$$

3.2.3 The Effect of Misalignment and Non-Orthogonality on the LLSE Algorithm

Misalignment and Non-Orthogonality errors have little effect on the algorithm performance when the misalignment of the y or z body axis rate gyros is solely in the y-z plane in the body axis. This is because the algorithm uses the change in amplitude much more than the actual value of that amplitude from the y and z axis rate gyros. However, whenever the misalignment of the y and z axis rate gyros permits some component of the shell spin rate to be detected by y or z axis gyros, a large bias is placed on the gyro data. To illustrate this, the simulation was run with gyro noise, GPS velocity noise, and gyro non-orthogonality active. In Figure 3.7 and Figure 3.8 , the x body axis was tilted 0.096 rad in the x-y plane and the y body axis gyro was tilted 0.096 rad into the x-y plane. Notice the bias in the y or "pitch" measured angular rate in Figure 3.7 . About 70 deg/sec of the 720 deg/sec spin rate is detected by the y body axis rate gyro. The x body axis detected angular rate error is harder to notice in Figure 3.7 but still significant. When this bias error and the incorrect shell spin rate estimate are introduced into the LLSE filter the error in roll estimate is quite large as shown in Figure 3.8 .

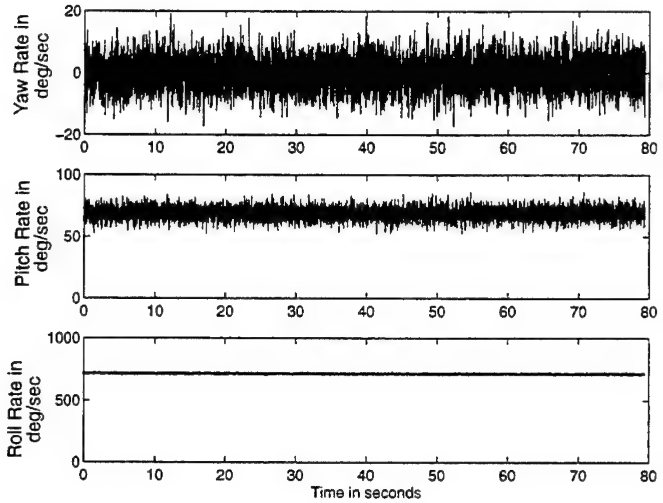


FIGURE 3.7 Measured Angular Rates in Body Axis with Non-Orthogonality of 0.096 rad in both X and Y Body Axis Rate Gyros in the X-Y Plane

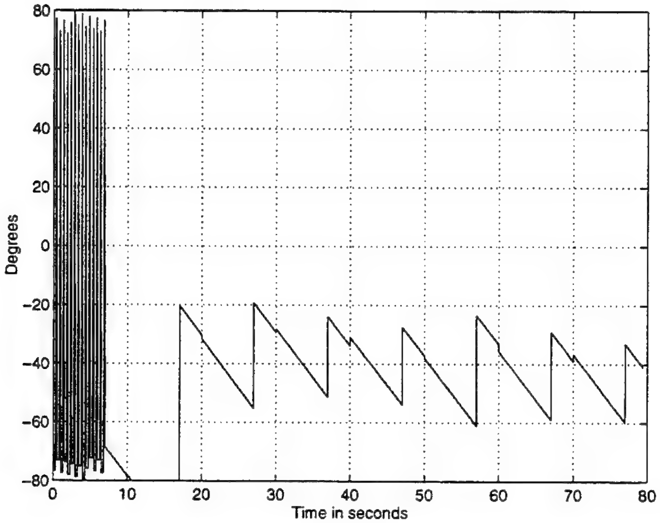


FIGURE 3.8 Roll Error with Non-Orthogonality of 0.096 rad in both X and Y Body Axis Rate Gyros in the X-Y Plane

In the next run, the z body axis was tilted 0.096 rad in the x-z plane in addition to the previous non-orthogonalities to indicate the decay in performance when both the y and z axis rate gyros detect a component of the shell spin rate. Figure 3.9 shows this effect on the algorithm roll estimation error when bias error from both the y and z axis rate gyros as well as the incorrect shell spin rate estimate are introduced into the LLSE filter.

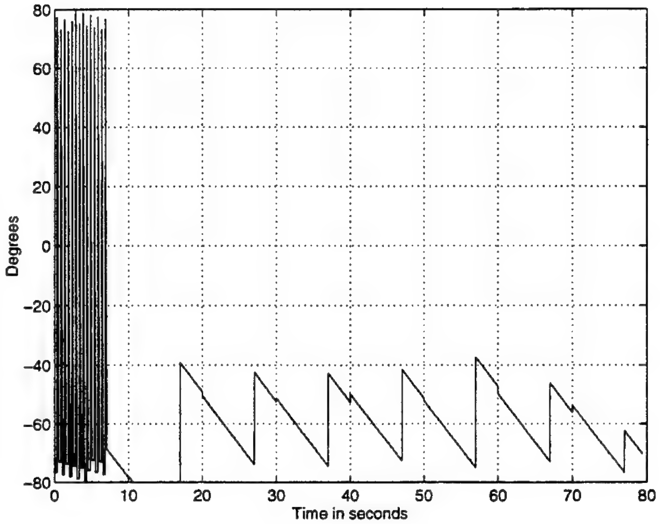


FIGURE 3.9 Roll Error with Non-Orthogonality of 0.096 rad in X, Y and Z Body Axis Rate Gyros

Because the bias error is a constant and the oscillatory y and z body axis rate gyro signal should be unbiased, estimating and removing this error is easily done by iteratively calculating the mean of the y and z body axis rate gyros over the batch processing time and subtracting this value from the subsequent rate gyro measured angular rates. This eliminates a significant portion of Misalignment and Non-Orthogonality error, leaving the incorrect shell spin rate error. The performance of the Y and Z axis bias error correction is shown below in Figure 3.10 .

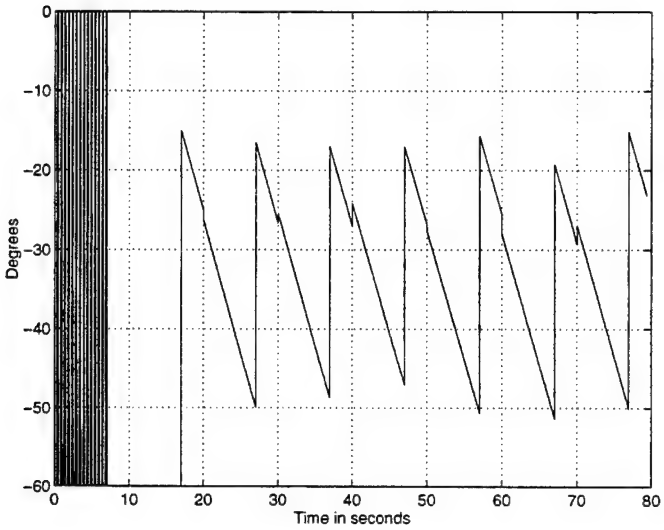


FIGURE 3.10 Roll Error with Non-Orthogonality of 0.096 rad in both X and Y Body Axis Rate Gyro in the X-Y Plane with Bias Error Correction Implemented

The best way to eliminate non-orthogonality error and MIMU misalignment error of the x axis rate gyro, is to calibrate the sensor carefully upon MIMU packaging and installation into the shell. This will insure the x axis rate gyro is lined up with the actual shell roll axis to a much better accuracy than the 0.096 rad (5.5 deg) used in this analysis. This correction will greatly reduce this error at roll rates of 2Hz or less.

3.2.4 Estimating Error in the X Body Axis Rate Gyro with the FFT

Combining the y and z axis rate gyro data as set up in Section 3.2 can provide a direct measurement of either sinϕ or cosϕ (sinϕ reshown below from Figure 3.3).

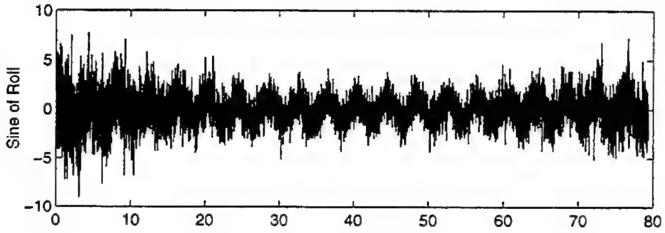


FIGURE 3.11 Sinϕ from y and z Axis Rate Gyros

Figure 3.11 indicates that the shell roll rate is the dominant frequency in this signal. An attempt was made to use the sinϕ measurements from the shell simulation with 2Hz spin rate (data at 1000Hz and micro gyros with noise as currently specified) to provide the roll rate of the shell independent of the x axis rate gyro.

3.2.4.1 N Dimensional Discrete Fourier Transform Description

An N Dimensional Discrete Fourier Transform (NDFT) is able to identify relative contributions of N discrete frequency components of a signal by utilizing orthogonality to separate a sampled function into individual contributions of each frequency. These frequency components are equally spaced between + and - the value of the sampling frequency divided by 2 (or $f_s/2$ where f_s = 1000Hz and so the frequencies are equally spaced between -500Hz and +500Hz). If the amount of sampled data is smaller than N (the number of frequencies components), the sampled data can be "zero padded" to fill the observation vector with 0's until it's size "n" is equal to N. A special form of the DFT (or algorithm to compute the DFT) is called the Fast Fourier Transform (FFT) and is especially useful in computer applications because of a reduced amount of calculations to provide the same results as the DFT. A complete discussion of the DFT and FFT is contained in reference [8] .

3.2.4.2 NFFT Performance as Roll Rate Estimator

Since the error we are trying to detect in the 2Hz roll rate is in the range of 1.44 deg/sec (0.004Hz), an NFFT will have to provide resolution greater than this to be of use. With a sampling frequency f_s = 1000Hz, an N = 1,500,000 point NFFT is used to provide discrete frequency component every 0.0006667Hz. Sampling sinϕ data for 10 seconds at 1000Hz produces a function vector $\overline{F}$ with 10,000 points. 10 seconds was chosen as the batch processing time to match the LLSE fast spinning algorithm Δt. This huge FFT requires 1,490,000 zeros to pad the function vector $\overline{F}$! A Matlab routine was created to perform the NFFT on data from the simulated 30km shell flight (79 seconds). The NFFT from one of the 10 second periods is shown below (scaled to the frequencies of interest).

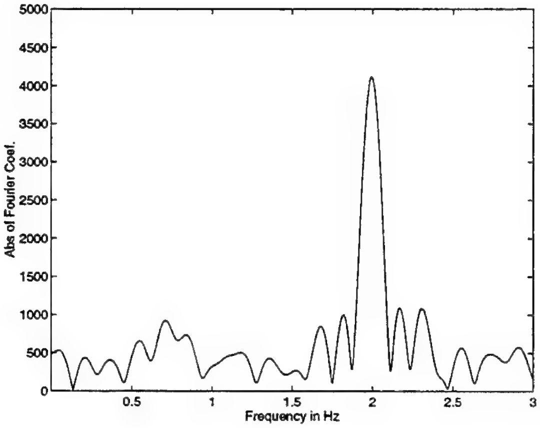


FIGURE 3.12 1,500,000 Point NFFT of 10s Sinϕ Data Sampled at 1000Hz

Extracting the maximum Fourier coefficient from this data placed the shell roll rate at 1.993Hz. The error in this particular estimate is 0.007Hz and is worse than the x axis rate gyro with scale factor error included. The results from the other 10 second periods of sampled data were similar in accuracy. The failure of the NFFT to predict the shell spin rate within required accuracy as demonstrated in this simulation, plus the need for a great deal of storage and processing capability in real time during the shell flight makes this approach undesirable. The performance of the NFFT at predicting roll rate is certain to get worse in the real environment where the roll rate is not exactly constant over the batch processing time period.

3.2.4.3 Estimating X Axis Gyro Scale Factor Error in the Navigation Filter

Another way must be found to estimate and remove the effects of scale factor error in the x axis gyro. Additional information from GPS position and velocity measurements as well as accelerometer information from the MIMU could be combined to provided an estimate of scale factor error. Remember that the final desired result of the Guided Munitions Demonstration Program is the creation of a complete navigation filter to estimate position, velocity and angular state. Current research [4] indicates this filter has the ability (once GPS is acquired) to estimate x axis scale factor error pretty well by utilizing information from the MIMU and GPS position. Prior to GPS acquisition, there is no other information available to estimate scale factor error. Since most of the control corrections will occur in the last half of the shell trajectory, 14 deg of error as an initial value is not too bad if the information to remove this error is available later in the flight.

3.3 The Fast Spinning Shell Algorithm without GPS

It would be nice not to wait for GPS data to begin the fast spinning algorithm to estimate the roll angle. This is true for a couple of reasons. First, remember the GMR time-line in Figure 1.2 which shows GPS will not be available until about 15-20 seconds into the flight of the shell. This does not leave much time for course corrections which cannot begin without some initial estimate of the local vertical state. If GPS was not necessary to estimate the local vertical state, then the time constraint is relaxed. The addition of GPS measurements at 20 seconds will simply improve the navigation solution. Second, the ballistic artillery shell could conceivably encounter jamming of its GPS receiver and lose access to GPS information. If this were to occur, independence from GPS measurements would allow the local vertical estimate to operate through the jamming.

Until now, the fast spinning roll estimating routine has used GPS velocity measurements to estimate the shell pitch over (gravity turn) rate "Q" and the shell yaw rate due to coriolis and repose forces "R." Since the azimuth and elevation of the gun barrel as well as the range of the trajectory are known a priori, good estimates of these values are available prior to GPS estimates of them. A crude demonstration yields some enlightening results. Let Q = -1 deg/sec (constant) and R = -0.01 deg/sec (constant) until GPS is available 27 seconds into the trajectory. Figure 3.13 shows the resulting Q and R estimates for the simulated flight.

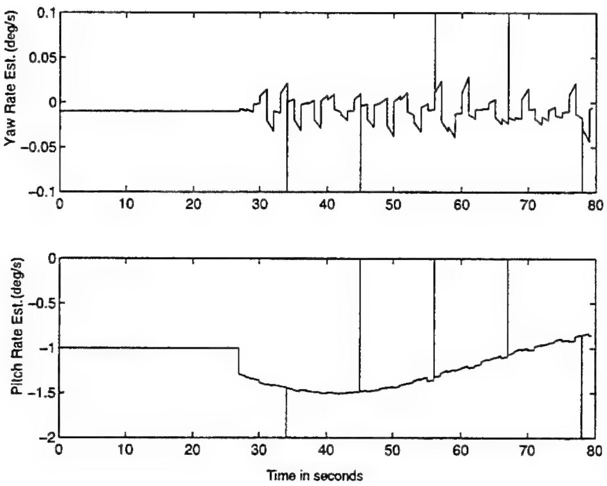


FIGURE 3.13 Q and R Estimates with No GPS for 27s

With the shell spin rate at 2Hz, the fast spinning algorithm performance in Figure 3.14 reveals no discernible improvement when GPS data becomes available.

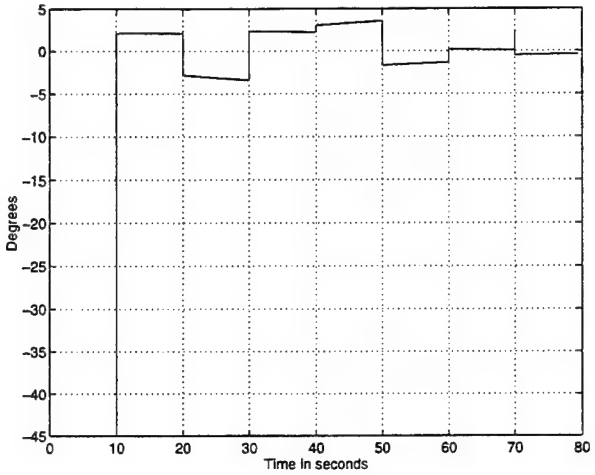


FIGURE 3.14 Roll Estimating Error No GPS for 27s

What would the roll estimation error look like without GPS information for the entire flight? The constant Q and R estimates are shown in Figure 3.15 and the resulting roll estimation error in shown in Figure 3.16 .

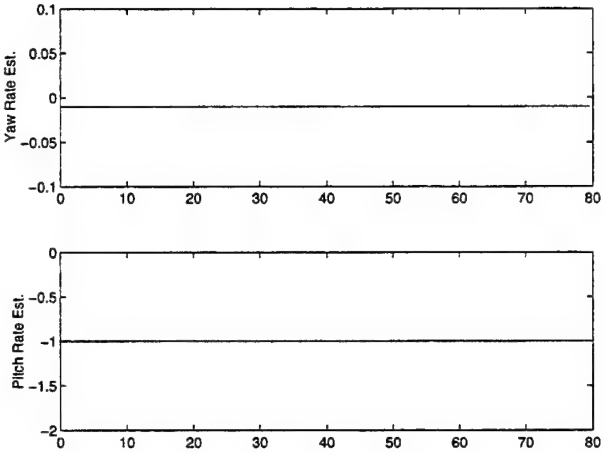


FIGURE 3.15 Q and R Estimates with No GPS

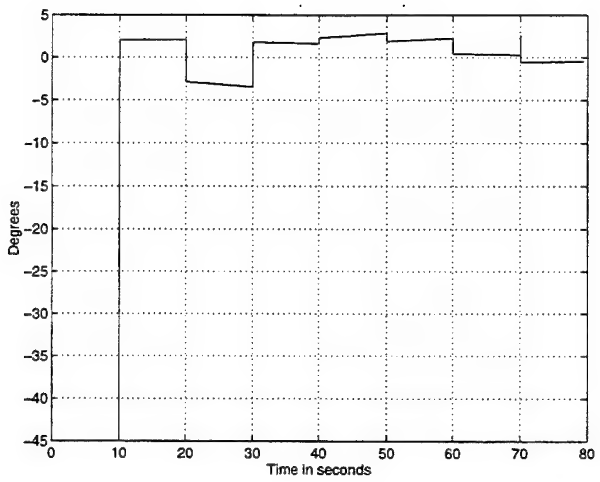


FIGURE 3.16 Fast Spinning Algorithm Roll Estimating Error No GPS

Notice that there is absolutely no difference between the algorithm performance without GPS in Figure 3.16 and the algorithm performance with GPS shown back in Figure 3.4 . This with simply constant estimates for the gravity turn of the shell! The fast spinning algorithm only uses the phase information between the y and the z axis rate gyro amplitudes, not the value of the amplitudes themselves. The shell pitch over rate Q and the shell yaw rate R will affect the observation noise as described later in Section 5.2.1.1; otherwise, the roll estimating algorithm performance is unaffected. In conclusion, this roll estimating routine is really an inertial only algorithm and has no need for GPS measurements.

Chapter 4

Estimating the Roll Angle of a Slowly Spinning Shell

The approach presented in Chapter 3 will only work if the shell is spinning because some oscillatory measurements are required to estimate the phase of the roll angle. Besides the non-spinning case, at very small roll rates the batch processing time employed by the "fast" spinning routine would have to be extremely long to include enough of the roll angle period to get a decent phase angle estimate. Clearly, another method is required for the slowly spinning shell case.

The material in this chapter will develop a different approach for the slowly spinning shell case. The first section will show how to use the pure lateral rate gyro measurements to estimate the roll angle of a non-spinning shell. Note that the problem must be solved without prior knowledge of the shell roll state. This method is different from the fast spinning method of Chapter 3 because preprocessing of the noisy rate gyro measurements is not required. This solution will resemble that of Chapter 3 because a similar deterministic linear least squares curve fitting method is used to deal with the rate gyro noise.

The slowly spinning problem is solved next in this chapter by simply taking the non-spinning solution and trying it in a simulation with a small roll rate. The errors that result will be discussed and corrected to complete the derivation of the slowly spinning shell roll estimating technique. The rest of the chapter is devoted to analysis of the new estimator performance in the presence of different rate gyro errors.

4.1 Solving the Non-Spinning Shell Problem

If the shell roll rate is negligible then the shell roll attitude is fixed at some unknown angle. Under these circumstances the job of extracting the roll angle from the angular rates measured in the body axis resembles the work done earlier for the spinning shell, with the exception that no sinusoidal oscillation of the x and y body rate gyro data will be present. EQ 3.1.1 and EQ 3.1.2 as first introduced in Chapter 3 (repeated below) represent the relationships necessary to extract the roll information from the body axis measured pitch and yaw rates.

$$\omega_y^{\ b} = Q\cos\phi + R\sin\phi \qquad \text{EQ (4.1.1)}$$

$$\omega_z^{\ b} = R\cos\phi - Q\sin\phi \qquad \text{EQ (4.1.2)}$$

We saw in the last sections that Q, R, $\omega_y^{\ b}$, and $\omega_z^{\ b}$ will be noisy measurements. In the non-spinning case the roll angle is constant and we have an independent set of linear equations in sinϕ and cosϕ that are not time varying.

4.1.1 Deterministic Linear Least Squares Roll Angle Estimation for the Non-Spinning Shell

EQ 4.1.1 and EQ 4.1.2 can be written in matrix form as shown below for samples taken at discrete time intervals t_1, t_2 .... t_n.

$$\begin{bmatrix} \omega_{y1}^{\;\;b} \\ \omega_{y2}^{\;\;b} \\ \omega_{y3}^{\;\;b} \\ \dots \\ \dots \\ \omega_{yn}^{\;\;b} \end{bmatrix} = \begin{bmatrix} Q_1 & R_1 \\ Q_2 & R_2 \\ Q_3 & R_3 \\ \dots & \dots \\ \dots & \dots \\ Q_n & R_n \end{bmatrix} \cdot \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \qquad \text{EQ (4.1.1.1)}$$

$$\begin{bmatrix} \omega_{z1}^{\;\;b} \\ \omega_{z2}^{\;\;b} \\ \omega_{z3}^{\;\;b} \\ \dots \\ \dots \\ \omega_{zn}^{\;\;b} \end{bmatrix} = \begin{bmatrix} R_1 & -Q_1 \\ R_2 & -Q_2 \\ R_3 & -Q_3 \\ \dots & \dots \\ \dots & \dots \\ R_n & -Q_n \end{bmatrix} \cdot \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \qquad \text{EQ (4.1.1.2)}$$

$\omega_{yi}^{\;b}$ is the y body axis rate gyro output

$\omega_{zi}^{\;b}$ is the z body axis rate gyro output

Q_i is the shell pitch over rate determined from GPS

R_i is the shell yaw rate determined from GPS

ϕ is the shell roll angle

These equations are in the form.

$$\bar{Y}_{nx1} = Z_{nx2} \cdot \bar{\alpha}_{2x1} \qquad \text{EQ (4.1.1.3)}$$

so that the linear least squares estimate of $\bar{\alpha}$ is given by the solution to the normal equations [8] (repeated below from Chapter 3). However, linear least squares estimation of

$$\bar{\alpha}_{LLSE} = \left(Z^T \cdot Z\right)^{-1} \cdot Z^T \cdot \bar{Y} \qquad \text{EQ (4.1.1.4)}$$

EQ 4.1.1 and EQ 4.1.2 separately (as set up in equations EQ 4.1.1.1 and EQ 4.1.1.2) produces large errors because the 2x2 matrix Z^TZ in the normal equations is ill-conditioned. This occurs because the non-spinning pitch rate “Q” is much larger than the non-spinning yaw rate “R.” Inverting the ill-conditioned matrix (very large eigenvalues) is nearly dividing by zero which magnifies estimation error. Since EQ 4.1.1 and EQ 4.1.2 are linearly independent, coupling data from both equations into one long Z matrix results in a properly conditioned Z^TZ matrix allowing the algorithm to yield decent results. The final linear least squares roll estimating algorithm takes the following form.

$$\begin{bmatrix} \omega_{y1}^{b} \\ \omega_{y2}^{b} \\ \omega_{y3}^{b} \\ \cdots \\ \cdots \\ \omega_{yn}^{b} \\ \omega_{z1}^{b} \\ \omega_{z2}^{b} \\ \omega_{z3}^{b} \\ \cdots \\ \cdots \\ \omega_{zn}^{b} \end{bmatrix} = \begin{bmatrix} Q_1 & R_1 \\ Q_2 & R_2 \\ Q_3 & R_3 \\ \cdots & \cdots \\ \cdots & \cdots \\ Q_n & R_n \\ R_1 & -Q_1 \\ R_2 & -Q_3 \\ R_3 & -Q_3 \\ \cdots & \cdots \\ \cdots & \cdots \\ R_n & -Q_n \end{bmatrix} \cdot \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \qquad \text{EQ (4.1.1.5)}$$

ω_{yi}^{b} is the y body axis rate gyro output

ω_{zi}^{b} is the z body axis rate gyro output

Q_i is the shell pitch over rate determined from GPS

R_i is the shell yaw rate determined from GPS

ϕ is the shell roll angle

or...

$$\bar{Y}_{nx1} = Z_{nx2} \cdot \bar{\alpha}_{2x1} \qquad \text{EQ (4.1.1.6)}$$

where...

$$\bar{\alpha}_{LLSE} = \left(Z^T \cdot Z\right)^{-1} \cdot Z^T \cdot \bar{Y} \qquad \text{EQ (4.1.1.7)}$$

4.1.2 Performance of the Linear Least Squares Non-Spinning Shell Roll Estimation Algorithm

A non-spinning shell trajectory was simulated at 1000Hz and the non-spinning shell algorithm developed above was coded into the Ada ballistic artillery shell simulation. The batch processing data quantization time for these results was 5 seconds (or 5000 data points at 1000Hz collected in $\overline{\mathrm{Y}}$ and Z for LLSE estimation). The roll error from this simulation in the absence of any sensor errors is given below in Figure 4.1 .

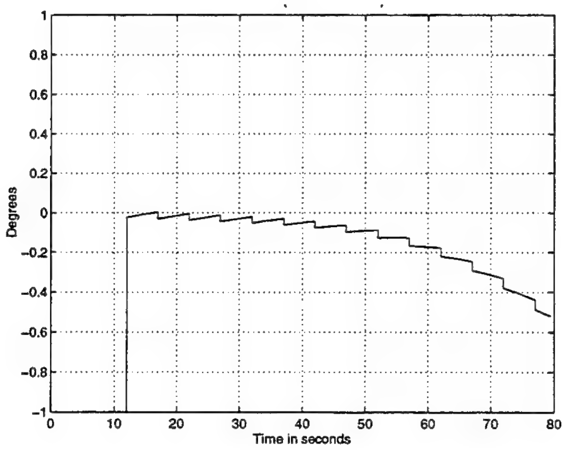


FIGURE 4.1 Non-Spinning Shell Roll Error (No Sensor Errors)

The simulation was rerun with the same parameters and micro gyro noise as specified in Chapter 6 and GPS velocity noise of $\sigma = 0.1$ m/s. The results are shown in Figure 4.2 .

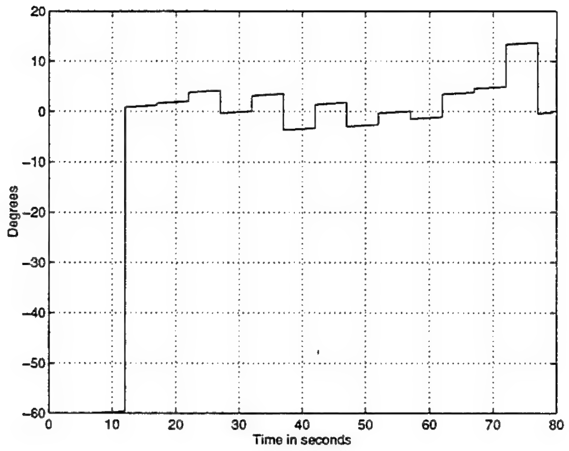


FIGURE 4.2 Non-Spinning Shell Roll Error (Noise Active)

4.2 Development of the Roll Estimating Algorithm when the Shell Spin Rate is Small

4.2.1 Trying the Non-Spinning Algorithm When the Shell Spin Rate is Small

The non-spinning shell formulas derived above apply over some finite time interval Δt if the shell roll rate P is assumed to be 0. The size of Δt and P will dictate the amount of error introduced by this assumption. If the shell spin rate is small the interval should be sufficiently large to provide decent estimates of the shell body axis pitch and yaw rates and yet be small enough to keep error due to the real roll rate acceptable. A preliminary slow spinning roll estimating algorithm emerges by first quantizing the shell trajectory into small time steps Δt where P is assumed to be 0. The algorithm was first tested with a shell spin rate of 10 degrees per second (0.02778Hz) and an initial shell roll angle equal 45 degrees. The system rate was 100Hz and the time interval was 1 second. This results in the number of observations per roll estimate N equal 100. The test did not include any sensor errors. The roll error from the algorithm is given in Figure 4.3 .

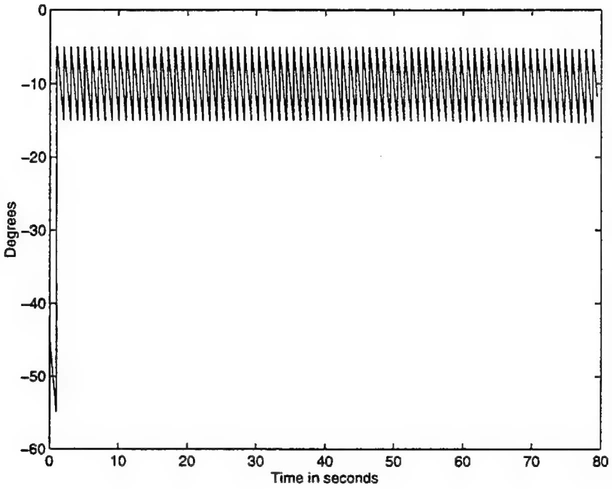


FIGURE 4.3 Slow Spin Rate Roll Error Δt = 1 sec

At the end of each time step the roll angle estimate is in error by 5 degrees. This 5 degree error is roughly constant throughout the trajectory and will be referred to as "bias" for the next few paragraphs (note: it looks like Bias = Spin Rate * Δt/2). From the initial 5 degree bias, the roll error increases linearly at the roll rate (10 deg/sec) across the time interval. This produces a ramp error that grows from the initial value of 5 degrees to a final value of 15 degrees over the one second interval. Another case was tested with the same trajectory and the time interval reduced to 0.2 seconds. At 100Hz the number of observations N per roll estimate was 20 for this case. Figure 4.4 shows the bias error reduced to about 2 degrees.

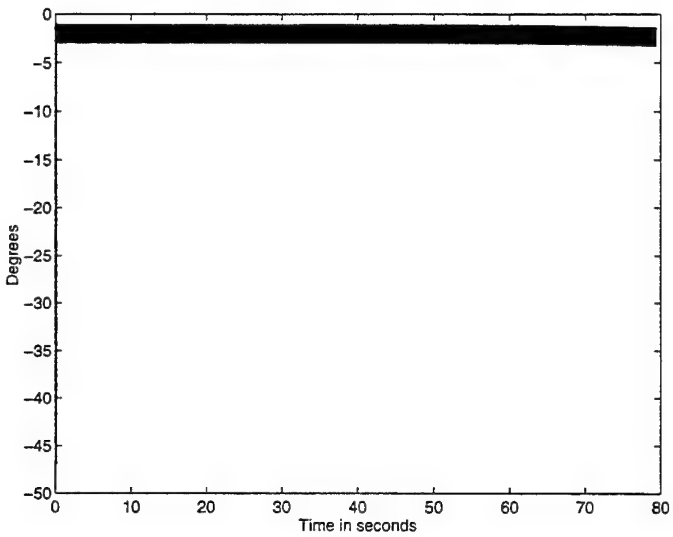


FIGURE 4.4 Slow Spin Rate Roll Error Δt = 0.2 sec

The algorithm accuracy improves as Δt gets smaller. However, as Δt gets smaller the algorithm becomes much more susceptible to noise. As will be shown a little later, the large amount of noise in the micro gyros will require Δt's greater than 4 or 5 seconds for reasonable results. Clearly, such a large Δt would introduce unacceptably large bias and roll rate errors.

4.2.2 Bias and Roll Rate Error Estimation for the Non-Spinning Shell Roll Algorithm

The linear least squares estimation fits a line through numerous data points to estimate sinϕ and cosϕ assuming the roll rate is zero during the batch processing time. Since roll rate is not zero for the slowly spinning shell case, the bias introduced by this assumption should be a function of the batch processing time and the actual shell roll rate. This is indeed the case. Observing the error bias for several simulation runs produced the following formula (hinted at earlier) for estimating the bias of the non-spinning algorithm roll error when operating in a slowly spinning trajectory.

$$Bias = \frac{P \cdot \Delta t}{2} \qquad \text{EQ (4.2.2.1)}$$

$$P = Shell\ Spin\ Rate$$

During the time to collect the next batch of data, the non-spinning algorithm assumes the roll angle is constant. The "saw tooth" effect in Figure 4.3 is the roll rate dynamic error building upon the already existing bias error because the shell is spinning slowly. This error can be eliminated through a two fold procedure. First, estimation of the shell roll rate (ω_x^b) from the x body axis rate gyro using the Maximum Likelihood technique from Appendix B and repeated below.

$$\hat{\omega}^b_x = \frac{\sum_{i=1}^{N} \omega^b_{xi}}{N} \qquad \text{EQ (4.2.2.2)}$$

Second, during the batch processing period, the roll rate estimate is used to increment the roll angle estimate from the LLSE value by the amount PΔt each discrete time step of the system. This relationship is given below.

$$\hat{\phi}_N = \hat{\phi}_{LLSE} + \hat{\omega}^b_x \cdot (N \cdot \Delta t) \qquad \text{EQ (4.2.2.3)}$$

N = Number of time steps since last LLSE estimate

4.2.3 Performance of the Final Slow Spinning Shell Roll Estimating Algorithm

Incorporating EQ 4.2.2.1 , EQ 4.2.2.2 , and EQ 4.2.2.3 into the non-spinning shell algorithm of Section 4.1 to remove the bias and roll rate dynamic errors yields the final form of the slow spinning roll angle estimation routine. The ballistic artillery shell simulation was run with a shell spinning at 15 degree/sec (0.04167Hz). The system rate was 1000Hz and the batch processing time for the slow spinning roll estimation algorithm was 5 seconds. The slow spinning algorithm performance without sensor errors is shown in Figure 4.5 .The simulation was rerun with the same parameters and micro gyro noise as specified in Chapter 6 and GPS velocity noise of $\sigma = 0.1$ m/s. The results are shown in Figure 4.6 .

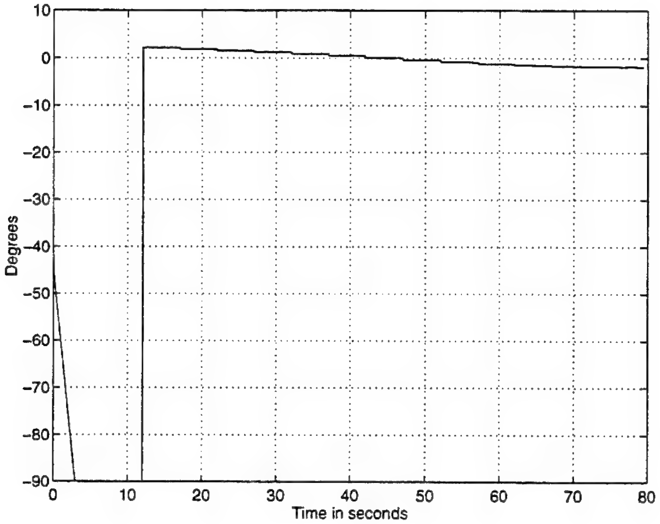


FIGURE 4.5 Final Slow Spin Algorithm Performance, P=0.04167Hz, System Rate = 1000Hz, (No Sensor Errors)

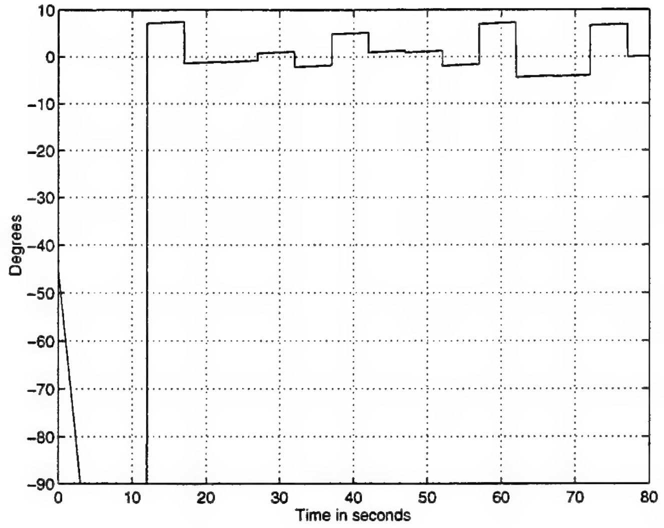


FIGURE 4.6 Final Slow Spin Algorithm Performance with Sensor Noise On, P=0.04167Hz, System Rate = 1000Hz

4.3 Selection of Batch Processing Time for the Slow Spin Roll Estimating Algorithm

Several batch processing intervals "Δt" were tested and the roll angle estimation performance was collected for each. The algorithm simulation incorporated the previous shell trajectory with shell spin rate P = 0.04167Hz. The system rate was 1000Hz and the micro gyro and GPS noise were active. The performance of 4 different LLSE batch processing Δt's is shown in Figure 4.7 .

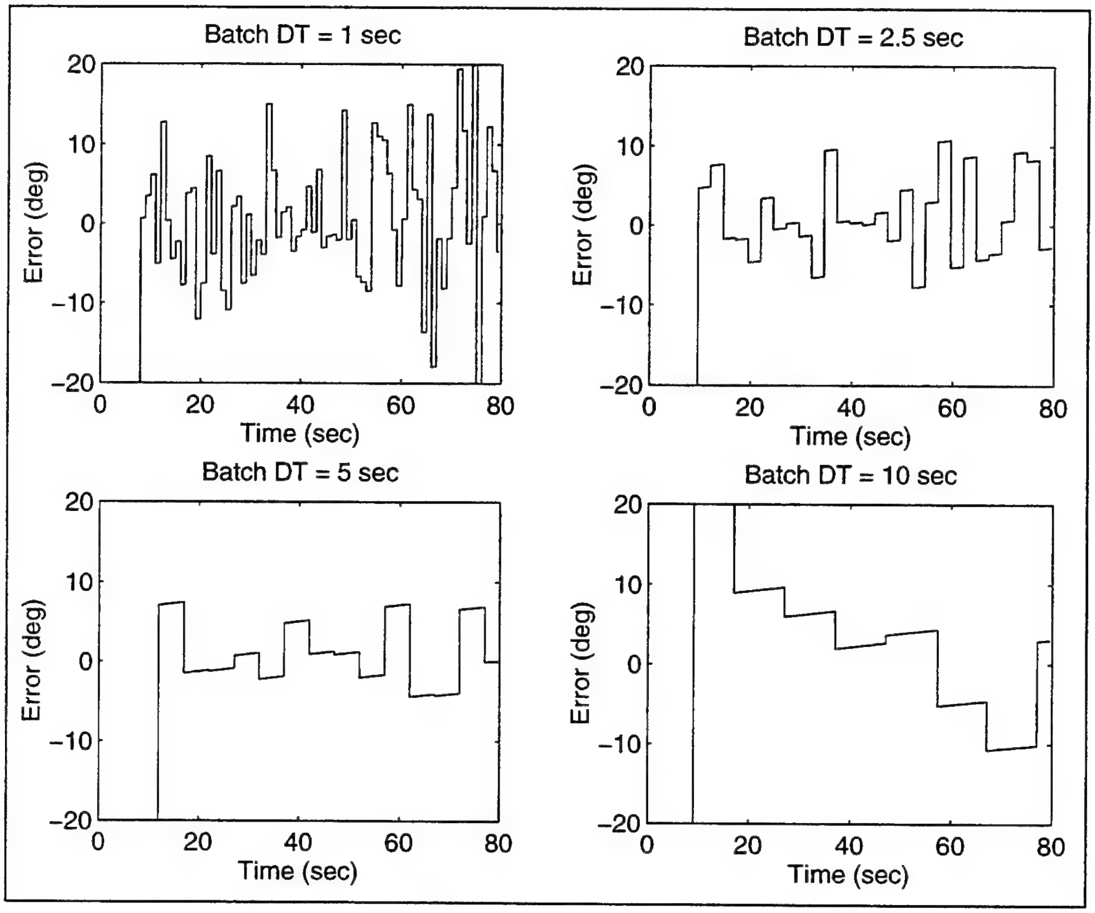


FIGURE 4.7 Slow Spin Roll Estimation Algorithm Performance for Differing Batch Processing Times (Δt)

Observation of the four examples shown suggests the choice of 5 seconds for the slow spinning algorithm batch processing Δt to produce the most accurate roll estimate.

4.4 Finding the Shell Roll Rate Crossover Between the Fast and Slow Spinning Roll Estimation Algorithms

The ballistic artillery shell simulation was coded to perform either the slow or the fast spinning roll estimating algorithms. The simulated micro gyro noise and GPS velocity noise was active for several test cases consisting of shell spin rates between 0 and 2

Hz. The goal of this exercise was to find the shell roll rate where the performance of both algorithms was roughly equal. This crossover point would be used in the overall estimator to select the appropriate algorithm depending on measured shell spin rate. Figure 4.9 shows the performance of both the slow spin and the fast spin algorithms side by side for the range of spin frequencies tested. This information is collected and present below in Figure 4.8 . The standard deviation and mean of each roll error plot shown in Figure 4.8 was calculated starting when the algorithm yields its first roll estimate. The absolute value of the bias and standard deviation were added to determine the performance crossover point.

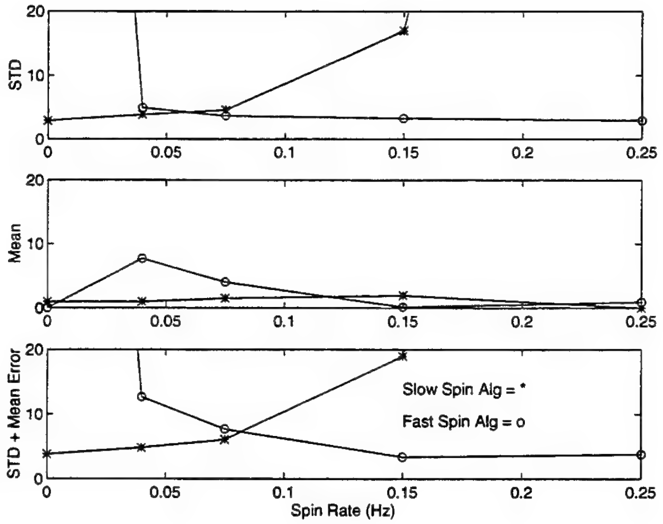


FIGURE 4.8 Fast and Slow Spin Roll Estimation Algorithm Performance Crossover Point

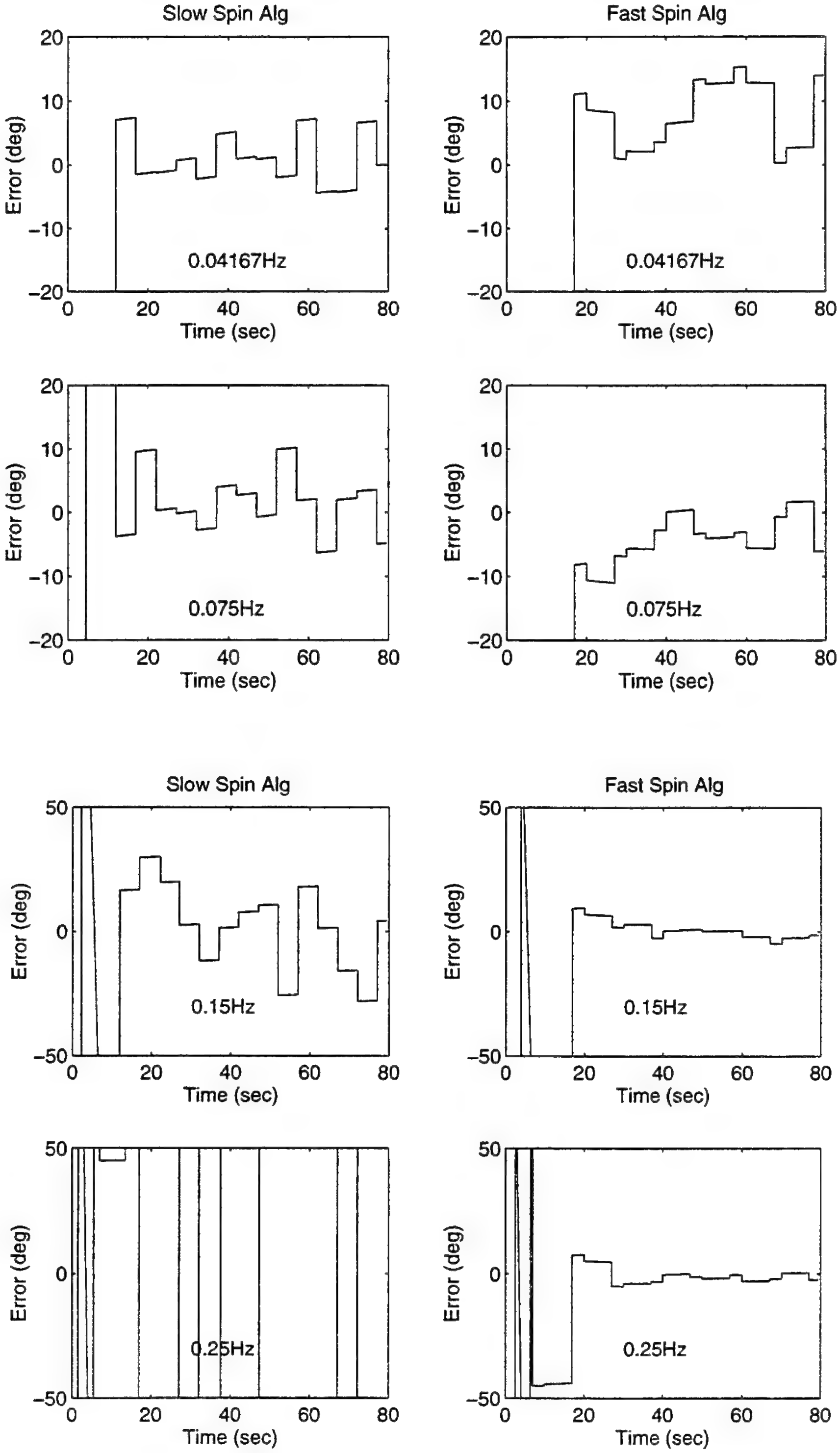


FIGURE 4.9 Comparison of Slow Spin and Fast Spin Algorithms for Several Shell Spin Rates

The fast spinning algorithm is more robust in frequency than the slow spinning routine. However, as the frequency approaches 0Hz the fast spinning algorithm develops a significant bias. The slow spinning algorithm is ineffective at roll rates where the period of the shell roll angle is less than the batch processing time Δt = 5 seconds. The performance cross over point is near 0.08Hz. This frequency represents the worst case performance of the filter incorporating both algorithms.

4.5 The Slow Spinning Shell Algorithm without GPS

How does the slow spinning algorithm perform without GPS during the entire flight? The simulation was run with 0.04Hz shell spin rate and with the shell pitch over rate estimate "Q" equal to a constant -1 deg/sec and the shell yaw rate estimate "R" equal to a constant -0.01 deg/sec. This was done for the fast spinning shell case in Section 3.3. The roll estimation error for this case in given below in Figure 4.10 .

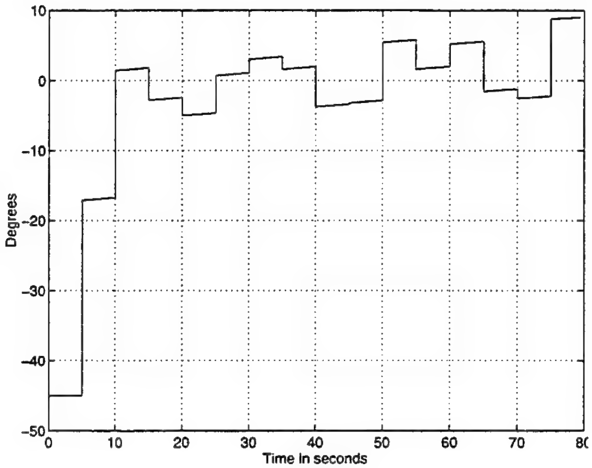


FIGURE 4.10 Slow Spinning Algorithm Roll Estimation Error with No GPS

The standard deviation of the no GPS roll estimation error is 3.9075deg. The roll estimation error with GPS shown in Figure 4.6 had a standard deviation of 4.6919deg. This is good news because, like the fast spinning shell algorithm, roll angle estimation does not need to wait for GPS measurements in the slow spinning case. In conclusion, the final roll estimating routine is an inertial only algorithm and has no need for GPS measurements.

Chapter 5

Kalman Filter Approach to Roll Angle Estimation

The goal of this chapter is to derive two Kalman filter implementations. I begin with an overview of Kalman filtering to familiarize the reader with the notation used throughout. The first Kalman filter will estimate the sine and cosine of the phase of the roll angle first introduced in Chapter 3 as the solution the "fast" spinning problem. This will be called the "fast spinning Kalman filter". The state space formulation of the Chapter 3 routine, the derivation of the associated error statistics, and analysis of the filter performance complete the first major section in the chapter.

The second part of the chapter develops a Kalman filter using the measurements and states introduced in Chapter 4 in solving the "slow" spinning problem. This will be called the "slow spinning Kalman filter". The state space formulation of the Chapter 4 routine, the derivation of the associated error statistics, and analysis of the filter performance complete the second major section in the chapter.

Finally, the fast and slow spinning Kalman filter's performances are compared at roll rates of 0Hz, 0.25Hz and 2Hz. Some interesting insights can be drawn which lead to the concluding chapter of the thesis.

5.1 Kalman Filter System Overview

The preceding sections have shown that the gyro measurements contain valuable attitude state information buried beneath noise and a predictable bias. Since the noise variance and bias can be characterized by understanding micro gyro behavior in differing dynamics and temperatures, this information can be used to remove the errors and uncover accurate and useful angular state information.

The Kalman filter is a recursive algorithm that provides a minimum variance estimate of the state errors through optimal use of the error characteristics. White noise of some variance is processed for each state in the Kalman filter. The discrete Kalman filter computes a state dynamics matrix relating the vehicle state vector $\overline{x}$ at times t+1 and t. This matrix is found by taking advantage of the state space relationship...

$$\frac{d}{dt}\bar{x}(t) = F(t)\cdot\bar{x}(t) + B(t)\cdot\bar{u}(t) + G(t)\cdot\bar{w}(t) \qquad \text{EQ (5.1.1)}$$

The finite difference equation relating discrete samples in the vector $\overline{x}$ at times t+1 and t in the absence of the forcing function is then...

$$\bar{x}(t+1) = A(t)\cdot\bar{x}(t) + w \qquad \text{EQ (5.1.2)}$$

The matrix A is a function of time and is updated every time step accordingly. The observations available from sensors are related to the state vector $\overline{x}$ by the following...

$$\bar{y}(t) = C(t)\cdot\bar{x}(t) + \bar{v} \qquad \text{EQ (5.1.3)}$$

The minimum error covariance linear least squares estimate for the state vector $\overline{x}$ based on the observations $\overline{y}$ is then calculated by the following recursive formulas...

$$\hat{\underline{x}}(t) = \hat{\underline{x}}(t|t-1) + \mathcal{P}(t|t-1)\,C(t)^T\left[C(t)\,\mathcal{P}(t|t-1)\,C(t)^T + \mathcal{R}(t)\right]^{-1}(\underline{y}(t) - C(t)\,\hat{\underline{x}}(t|t-1)) \qquad \text{EQ (5.1.4)}$$

$$\mathcal{P}(t) = \mathcal{P}(t|t-1) - \mathcal{P}(t|t-1)\,C(t)^T\left[C(t)\,\mathcal{P}(t|t-1)\,C(t)^T + \mathcal{R}(t)\right]^{-1} C(\,(t)\,\mathcal{P}(t|t-1)\,) \qquad \text{EQ (5.1.5)}$$

The error covariances of the state vector "$\mathcal{P}$' and the noise sequence v, along with the strength of the white noise w are as follows...

$$\underline{\tilde{x}} = \underline{x} - \underline{\hat{x}} \qquad \text{EQ (5.1.6)}$$

$$\mathcal{P} = E\left[\underline{\tilde{x}} \cdot \underline{\tilde{x}}^T\right] \qquad \text{EQ (5.1.7)}$$

$$E\left[\underline{w}(t) \cdot \underline{w}(t)^T\right] = Q(t) \cdot \delta(t-\tau) \qquad \text{EQ (5.1.8)}$$

$$E\left[\underline{v}(t_i) \cdot \underline{v}(t_j)^T\right] = \begin{pmatrix} \mathcal{R}(t_i) \sim i = j \\ 0 \sim i \neq j \end{pmatrix} \qquad \text{EQ (5.1.9)}$$

The state vector $\overline{\mathrm{x}}$ and the error covariance matrix $\mathcal{P}$(t) are propagated every time step using A(t) and the covariance matrix of the discrete noise vector $\overline{\mathrm{w}}$, denoted as Q. The propagation equations are as follows.

$$\underline{\hat{x}}(t+1|t) = A(t) \cdot \underline{\hat{x}}(t) \qquad \text{EQ (5.1.10)}$$

$$\mathcal{P}(t+1|t) = A(t) \cdot \mathcal{P}(t) \cdot A(t)^T + G(t) \cdot Q(t) \cdot G(t)^T \qquad \text{EQ (5.1.11)}$$

A complete treatment of Kalman filtering can be found in reference [12] . This reference was the source of the information presented in this section and the notation from this reference was used throughout. (Note: t = t_0, t_0+1..., where 1 is the discrete time increment, not necessarily 1 second) A block diagram of this filter is given in Figure 5.1 .

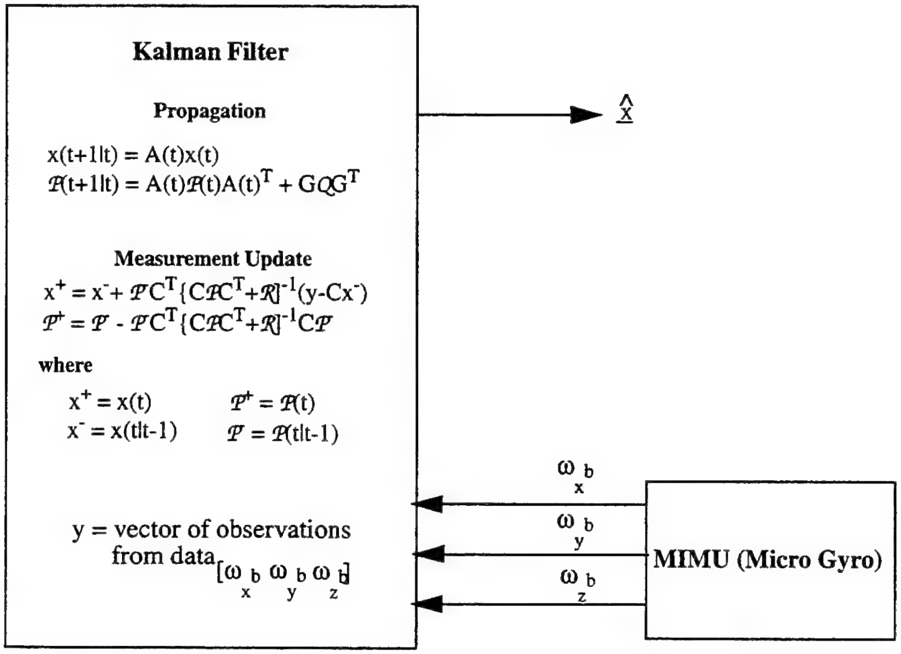

Kalman Filter
Propagation
x(t+1|t) = A(t)x(t)
𝒫(t+1|t) = A(t)𝒫(t)A(t)^T + G𝒬G^T
Measurement Update
x+ = x- + 𝒫C^T{C𝒫C^T+ℛ}^-1(y-Cx-)
𝒫+ = 𝒫 - 𝒫C^T{C𝒫C^T+ℛ}^-1C𝒫
where
x+ = x(t)
𝒫+ = 𝒫(t)
x- = x(t|t-1)
𝒫 = 𝒫(t|t-1)
y = vector of observations from data [ω b x ω b y ω b z]
x̂
ω b x
ω b y
ω b z
MIMU (Micro Gyro)

FIGURE 5.1 Kalman Filter Implementation

5.2 Kalman Filter Implementation of Fast Spinning Algorithm

5.2.1 State Space Realization for Algorithm

A five state Kalman Filter was developed for the fast spinning shell case to provide the iterative optimal estimate of $\cos\Phi_P$ and $\sin\Phi_P$ originally introduced in Section 3.2 on page 35 for the fast spinning LLSE batch processing technique. The other three states used include the shell roll rate “P”, and the y and z body axis lumped equivalent rate gyro biases, “Bias_y” and “Bias_z”. “Lumped equivalent rate gyro biases” are bias errors in the rate gyro angular rate measurements due to some combination of rate gyro non-orthogonality error, scale factor error and bias stability error. The dynamics of each state are modelled as a constant plus noise as shown in EQ 5.2.1.1 . Note: no process noise "*Q*" will be used in the filter.

$$\underline{x}(t+1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \underline{x}(t) + \underline{w} \qquad \text{EQ (5.2.1.1)}$$

The observations of the gyro biases come from preprocessing the y and z body axis rate gyro data. This preprocessing simply calculates the recursive average of the gyro signals shown below in EQ 5.2.1.2 and EQ 5.2.1.3 to use as measurements in the filter.

$$YGyroLumpedBias_{Observation} = \sum \frac{\omega_y^{\;b}}{n} \qquad \text{EQ (5.2.1.2)}$$

$$ZGyroLumpedBias_{Observation} = \sum \frac{\omega_z^{\;b}}{n} \qquad \text{EQ (5.2.1.3)}$$

$\omega_y^{\ b}$ is the y body axis rate gyro output
$\omega_z^{\ b}$ is the z body axis rate gyro output
n is the number of discrete time steps elapsed

The relation between the rate gyro lumped equivalent bias and the observations of the states cosΦ_P and sinΦ_P is shown below. Because these four states all depend on the y and z axis rate gyro outputs, the measurements are correlated with each other. This correlation decrease with time. The measurement covariance matrix is derived in Section 5.2.1.1.

$$\sin\phi\,(t) \;=\; \frac{R\omega_y^{\;b} - Q\omega_z^{\;b}}{Q^2 + R^2} \qquad \text{EQ (5.2.1.4)}$$

$$\sin\phi\,(t) \;=\; \sin\,(P\cdot t)\,\cdot\,\cos\Phi_P + \cos\,(P\cdot t)\,\cdot\,\sin\Phi_P \qquad \text{EQ (5.2.1.5)}$$

which leads to the following for sinφ(t)...

$$\sin\phi\,(t) \;=\; \omega_{bias}^{\;\;y}\cdot\left(\frac{R}{Q^2+R^2}\right) - \omega_{bias}^{\;\;z}\cdot\left(\frac{Q}{Q^2+R^2}\right) + \sin\,(P\cdot t)\,\cdot\,\cos\Phi_P + \cos\,(P\cdot t)\,\cdot\,\sin\Phi_P \qquad \text{EQ (5.2.1.6)}$$

and the similar relationship for cosφ(t) is...

$$\cos\phi\,(t) \;=\; \frac{Q\omega_y^{\;b} + R\omega_z^{\;b}}{Q^2 + R^2} \qquad \text{EQ (5.2.1.7)}$$

$$\cos\phi\,(t) \;=\; \omega_{bias}^{\;\;y}\cdot\left(\frac{Q}{Q^2+R^2}\right) + \omega_{bias}^{\;\;z}\cdot\left(\frac{R}{Q^2+R^2}\right) + \cos\,(P\cdot t)\,\cdot\,\cos\Phi_P - \sin\,(P\cdot t)\,\cdot\,\sin\Phi_P \qquad \text{EQ (5.2.1.8)}$$

The final state space realization for the fast spinning shell case Kalman Filter is summarized below.

$$\begin{bmatrix} \omega_x^{\;b} \\ \sum\frac{\omega_y^{\;b}}{n} \\ \sum\frac{\omega_z^{\;b}}{n} \\ \sin\phi\,(t) \\ \cos\phi\,(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{R}{Q^2+R^2} & \frac{-Q}{Q^2+R^2} & \sin\,(P\cdot t) & \cos\,(P\cdot t) \\ 0 & \frac{Q}{Q^2+R^2} & \frac{R}{Q^2+R^2} & \cos\,(P\cdot t) & -\sin\,(P\cdot t) \end{bmatrix} \cdot \begin{bmatrix} P \\ \omega_{bias}^{\;\;y} \\ \omega_{bias}^{\;\;z} \\ \cos\Phi_P \\ \sin\Phi_P \end{bmatrix} + \underline{v} \qquad \text{EQ (5.2.1.9)}$$

where...

$\omega_x^{\;b}$ is the x body axis rate gyro output

$\Sigma\omega_y^{\;b}/n$ is the average y body axis rate gyro output

$\Sigma\omega_z^{\;b}/n$ is the average z body axis rate gyro output

sinφ(t) is the measurement created via EQ 3.1.2

cosϕ(t) is the measurement created via EQ 3.2.1
P is the shell roll rate estimate
Q is the average shell pitch over rate determined priori
R is the average shell yaw rate determined priori
$\omega_{\text{bias}}{}^{y}$ is the estimate of the lumped average y body rate gyro bias
$\omega_{\text{bias}}{}^{z}$ is the estimate of the lumped average z body rate gyro bias
Φ_P is the phase of the shell roll angle from EQ 3.2.3

which is of the form...

$$\underline{y}(t) = C(t) \cdot \underline{x}(t) + \underline{v} \qquad \text{EQ (5.2.1.10)}$$

and

$$\underline{x}(t+1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \underline{x}(t) + \underline{w} \qquad \text{EQ (5.2.1.11)}$$

which is of the form...

$$\underline{x}(t+1) = A(t) \cdot x(t) + \underline{w} \qquad \text{EQ (5.2.1.12)}$$

5.2.1.1 Error Modelling of Observation Noise

The covariance matrix of the observation noise "$\mathcal{R}$" is shown in EQ 5.2.1.1.1 .Notice that "$\mathcal{R}$" is nearly singular when t/Δt equals one. This problem is avoided by starting the filter when t/Δt equals two.

$$\mathcal{R}(t) = \begin{bmatrix} 0.007162 & 0 & 0 & 0 & 0 \\ 0 & \frac{0.007162\Delta t}{t} & 0 & \frac{0.00406\Delta t}{t} & \frac{0.40628\Delta t}{t} \\ 0 & 0 & \frac{0.007162\Delta t}{t} & -\frac{0.40628\Delta t}{t} & \frac{0.00406\Delta t}{t} \\ 0 & \frac{0.00406\Delta t}{t} & -\frac{0.40628\Delta t}{t} & 23.05 & 0 \\ 0 & \frac{0.40628\Delta t}{t} & \frac{0.00406\Delta t}{t} & 0 & 23.05 \end{bmatrix} \qquad \text{EQ (5.2.1.1.1)}$$

The elements of the covariance matrix are derived below. To begin, the micro gyro angle random walk of 9.2 deg/rt-hr yields the rate error variance as described in EQ 5.2.1.1.2 .

$$cov\left(\omega_x^{\,b}\right) = \left(\left(9.2 \cdot \frac{deg}{\sqrt{hr}}\right) \cdot \frac{\pi \cdot rad}{180 \cdot deg}\right)^2 \cdot \frac{1 hour}{3600 s} \cdot \frac{1000}{s} = 0.007162 \cdot \frac{rad^2}{s^2} \qquad \text{EQ (5.2.1.1.2)}$$

$$cov\left(\omega_x^{\,b}\right) = cov\left(\omega_y^{\,b}\right) = cov\left(\omega_z^{\,b}\right) \qquad \text{EQ (5.2.1.1.3)}$$

Note that the noise for the bias state observations is time varying as shown in EQ 5.2.1.1.4 .

$$cov\left(\frac{\sum \omega_n}{n}\right) = \frac{1}{n^2} \cdot cov\left(\sum \omega_n\right) = \frac{cov\left(\omega_n\right)}{n} \qquad \text{EQ (5.2.1.1.4)}$$

The relationship between the covariance of the sinϕ(t) and cosϕ(t) (observations 4 and 5) and the shell pitch over rate "Q" and yaw rate "R" is developed below.

$$E\left[\sin\phi\,(t) \cdot \sin\phi\,(t)\right] = E\left[\left(\frac{R\omega_y^{\,b} - Q\omega_z^{\,b}}{Q^2 + R^2}\right)^2\right] = \mathcal{R}(4, 4) \qquad \text{EQ (5.2.1.1.5)}$$

$$E\left[\cos\phi\,(t) \cdot \cos\phi\,(t)\right] = E\left[\left(\frac{Q\omega_y^{\,b} + R\omega_z^{\,b}}{Q^2 + R^2}\right)^2\right] = \mathcal{R}(5, 5) \qquad \text{EQ (5.2.1.1.6)}$$

$$\mathcal{R}(4, 4) = \left(\frac{R}{Q^2 + R^2}\right)^2 \cdot E\left[\omega_y^{\,b} \cdot \omega_y^{\,b}\right] + \left(\frac{Q}{Q^2 + R^2}\right)^2 \cdot E\left[\omega_z^{\,b} \cdot \omega_z^{\,b}\right] \qquad \text{EQ (5.2.1.1.7)}$$

$$\mathcal{R}(5, 5) = \left(\frac{Q}{Q^2 + R^2}\right)^2 \cdot E\left[\omega_y^{\,b} \cdot \omega_y^{\,b}\right] + \left(\frac{R}{Q^2 + R^2}\right)^2 \cdot E\left[\omega_z^{\,b} \cdot \omega_z^{\,b}\right] \qquad \text{EQ (5.2.1.1.8)}$$

The covariance of sinϕ(t) and cosϕ(t) will change with time in magnitude if Q and R vary. Note that when both Q and R are small in magnitude, the covariance can become very large. The simulation is run using constant Q = -0.017453 rad/sec and R = -0.0017453 rad/sec which results in the following...

$$\mathcal{R}(4,4) = 0.321812 \cdot Var\left(\omega_y{}^b\right) + 3218.12 \cdot Var\left(\omega_z{}^b\right) = 23.05 \qquad \text{EQ (5.2.1.1.9)}$$

$$\mathcal{R}(5,5) = 3218.12 \cdot Var\left(\omega_y{}^b\right) + 0.321812 \cdot Var\left(\omega_z{}^b\right) = 23.05 \qquad \text{EQ (5.2.1.1.10)}$$

Note: 23.05 is a large value for the variance of the error in the value of sine and cosine. This large number is due to a combination of the constants Q and R and the large rate gyro noise. Assuming the y and z axis rate gyro signals are corrupted by independent zero mean white noise processes, the following relationships are true for the cross covariance of sinϕ(t) and cosϕ(t).

$$E\left[\omega_y{}^b \cdot \omega_z{}^b\right] = 0 \qquad \text{EQ (5.2.1.1.11)}$$

$$E\left[\cos\phi(t) \cdot \sin\phi(t)\right] = \frac{R \cdot Q}{\left(Q^2+R^2\right)^2} \cdot Var\left(\omega_y{}^b\right) - \frac{R \cdot Q}{\left(Q^2+R^2\right)^2} \cdot Var\left(\omega_z{}^b\right) = 0 \qquad \text{EQ (5.2.1.1.12)}$$

The cross correlation between observations $\Sigma\omega_y{}^b/n$, $\Sigma\omega_z{}^b/n$, sinϕ(t) and cosϕ(t) are developed below.

$$E\left[\omega_z{}^b \cdot \sum \frac{\omega_y}{n}^b\right] = E\left[\omega_y{}^b \cdot \sum \frac{\omega_z}{n}^b\right] = 0 \qquad \text{EQ (5.2.1.1.13)}$$

The relationship in EQ 5.2.1.1.14 is true because the rate gyro measurements at the present time are uncorrelated with all the rate gyro measurements in the past.

$$E\left[\omega_y{}^b \cdot \sum \frac{\omega_y}{n}^b\right] = E\frac{\left[\omega_y{}^b \cdot \omega_y{}^b\right]}{n} = E\left[\omega_z{}^b \cdot \sum \frac{\omega_z}{n}^b\right] = E\frac{\left[\omega_z{}^b \cdot \omega_z{}^b\right]}{n} = \frac{0.007162}{n} \qquad \text{EQ (5.2.1.1.14)}$$

Which leads to...

$$E\left[\sin\phi(t) \cdot \sum \frac{\omega_y}{n}^b\right] = E\left[\left(\frac{R\omega_y{}^b - Q\omega_z{}^b}{Q^2+R^2}\right) \cdot \sum \frac{\omega_y}{n}^b\right] = \mathcal{R}(2,4) = \mathcal{R}(4,2) \qquad \text{EQ (5.2.1.1.15)}$$

$$\mathcal{R}(2,4) = \mathcal{R}(4,2) = 0.5672E\left[\omega_y{}^b \cdot \sum \frac{\omega_y}{n}^b\right] - 56.728E\left[\omega_z{}^b \cdot \sum \frac{\omega_y}{n}^b\right] = \frac{0.00406}{n} \qquad \text{EQ (5.2.1.1.16)}$$

$$E\left[\sin\phi(t)\cdot\sum\frac{\omega_z^{\,b}}{n}\right] = E\left[\left(\frac{R\omega_y^{\,b}-Q\omega_z^{\,b}}{Q^2+R^2}\right)\cdot\sum\frac{\omega_z^{\,b}}{n}\right] = \mathcal{R}(3,4) = \mathcal{R}(4,3) \qquad \text{EQ (5.2.1.1.17)}$$

$$\mathcal{R}(3,4) = \mathcal{R}(4,3) = 0.5672E\left[\omega_y^{\,b}\cdot\sum\frac{\omega_z^{\,b}}{n}\right] - 56.728E\left[\omega_z^{\,b}\cdot\sum\frac{\omega_z^{\,b}}{n}\right] = -\frac{0.40628}{n} \qquad \text{EQ (5.2.1.1.18)}$$

$$E\left[\cos\phi(t)\cdot\sum\frac{\omega_y^{\,b}}{n}\right] = E\left[\left(\frac{Q\omega_y^{\,b}+R\omega_z^{\,b}}{Q^2+R^2}\right)\cdot\sum\frac{\omega_y^{\,b}}{n}\right] = \mathcal{R}(2,5) = \mathcal{R}(5,2) \qquad \text{EQ (5.2.1.1.19)}$$

$$\mathcal{R}(2,5) = \mathcal{R}(5,2) = 56.728E\left[\omega_y^{\,b}\cdot\sum\frac{\omega_y^{\,b}}{n}\right] + 0.5672E\left[\omega_z^{\,b}\cdot\sum\frac{\omega_y^{\,b}}{n}\right] = \frac{0.40628}{n} \qquad \text{EQ (5.2.1.1.20)}$$

$$E\left[\cos\phi(t)\cdot\sum\frac{\omega_z^{\,b}}{n}\right] = E\left[\left(\frac{Q\omega_y^{\,b}+R\omega_z^{\,b}}{Q^2+R^2}\right)\cdot\sum\frac{\omega_z^{\,b}}{n}\right] = \mathcal{R}(3,5) = \mathcal{R}(5,3) \qquad \text{EQ (5.2.1.1.21)}$$

$$\mathcal{R}(3,5) = \mathcal{R}(5,3) = 56.728E\left[\omega_y^{\,b}\cdot\sum\frac{\omega_z^{\,b}}{n}\right] + 0.5672E\left[\omega_z^{\,b}\cdot\sum\frac{\omega_z^{\,b}}{n}\right] = \frac{0.00406}{n} \qquad \text{EQ (5.2.1.1.22)}$$

5.2.1.2 Initial State and Initial Error Covariance Matrix

To initialize the Kalman Filter, an initial guess at each state variable is required "$\bar{x}_0$", and the error covariance matrix associated with the initial guess or "$\mathcal{P}_0$". The first three states are readily initialized by the first value of the appropriate micro gyro output. The corresponding initial error covariance is simply the covariance of the micro gyro output in "$\mathcal{R}(1,1)$". Since $\cos\Phi_P$ and $\sin\Phi_P$ are unknown, an initial value $\Phi_P = 0$ deg is assumed. The initial error covariance for $\cos\Phi_P$ and $\sin\Phi_P$ is found by first assuming Φ_P is distributed uniformly between 0 and 2π. The expected values of both $\cos\Phi_P$ and $\sin\Phi_P$ are derived in EQ 5.2.1.2.1 and EQ 5.2.1.2.2 and represents the initial error covariance for these states.

$$E[(\cos x - 1)^2] = \int_{-\pi}^{\pi}\frac{(\cos x - 1)^2}{2\pi}dx = 1.5 \qquad \text{EQ (5.2.1.2.1)}$$

$$E[\sin x^2] = \int_{-\pi}^{\pi} \frac{\sin x^2}{2\pi} dx = 0.5$$ EQ (5.2.1.2.2)

$$\bar{x}_0 = \begin{bmatrix} \omega_x^{\ b}(t_0) \\ \omega_y^{\ b}(t_0) \\ \omega_z^{\ b}(t_0) \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} P(t_0) \\ \omega_{bias}^{\ y}(t_0) \\ \omega_{bias}^{\ z}(t_0) \\ \cos\Phi_{P0} \\ \sin\Phi_{P0} \end{bmatrix}$$ EQ (5.2.1.2.3)

$$\mathcal{P}_0 = \begin{bmatrix} 0.007162 & 0 & 0 & 0 & 0 \\ 0 & 0.007162 & 0 & 0 & 0 \\ 0 & 0 & 0.007162 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$ EQ (5.2.1.2.4)

This state space realization is used along with the Kalman Filter equations described in Section 5.1 to provide the optimal state estimate. Every time step cosΦ_P, sinΦ_P, and P are extracted and used as shown below to determine the roll estimate for the shell. The matrix C(t) is also updated every time step with the new “P” estimate.

$$\sin\hat{\phi}(t) = \sin(P \cdot t + \Phi_P)$$ EQ (5.2.1.2.5)

$$\cos\hat{\phi}(t) = \cos(P \cdot t + \Phi_P)$$ EQ (5.2.1.2.6)

$$\hat{\phi}(t) = \operatorname{atan}\left(\frac{\sin\hat{\phi}(t)}{\cos\hat{\phi}(t)}\right)$$ EQ (5.2.1.2.7)

5.2.2 Kalman Filter Performance

This Kalman Filter was coded into the Ada simulation for Monte Carlo analysis. The trajectory modelled is the same 30Km 155mm howitzer simulation described in Section 6.3 with a 2Hz shell roll rate. GPS measurements were not used for any of these simulations. The shell pitch over rate “Q” was set to a constant value of -1 deg/sec and the shell yaw rate “R” was set to a constant value of -0.01 deg/sec. The Kalman Filter performance in relation to the modelled micro gyro errors specified in Section 6.1 is presented below.

5.2.2.1 Performance in Noise Only

The Kalman Filter roll error in the noise only environment is given in Figure 5.2 . The y and z axis rate gyro non-orthogonality was also active to demonstrate the filters ability to estimate and remove bias. Roll error standard deviation for this case was a mere 0.7098 degrees.

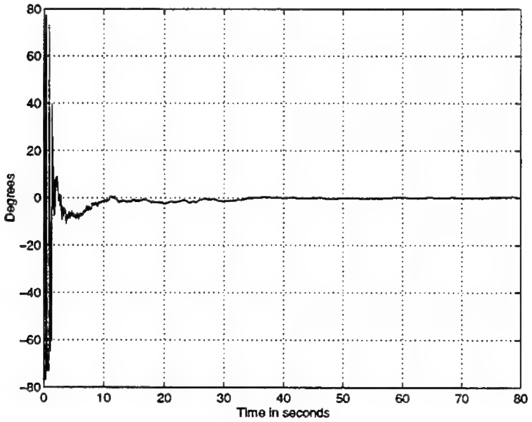


FIGURE 5.2 Fast Spin Kalman Filter Roll Error Noise Only Case

The performance of the individual states in the Kalman Filter are given in Figure 5.3 , Figure 5.4 , Figure 5.5 , Figure 5.6 , and Figure 5.7 . The phase estimate from states four and five is provided in Figure 5.8 .

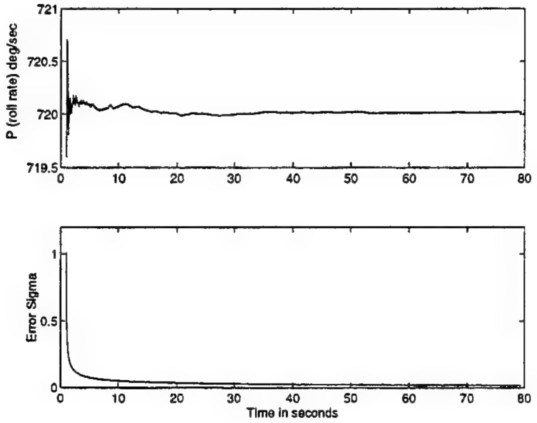


FIGURE 5.3 Roll Rate Estimate from Kalman Filter

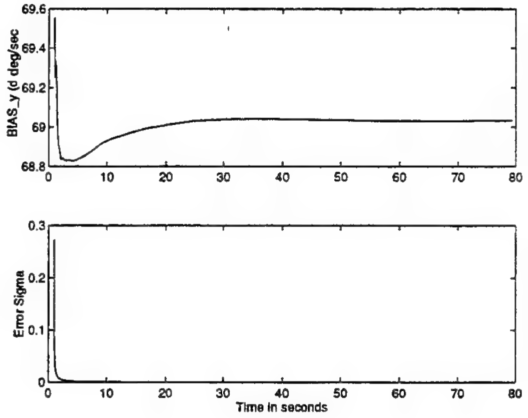


FIGURE 5.4 Y Body Axis Rate Gyro Bias Estimate from Kalman Filter

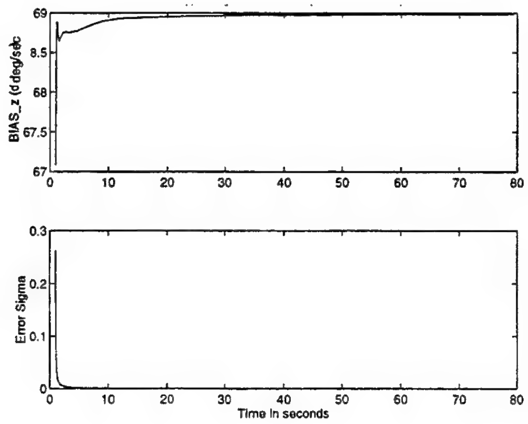


FIGURE 5.5 Z Body Axis Rate Gyro Bias Estimate from Kalman Filter

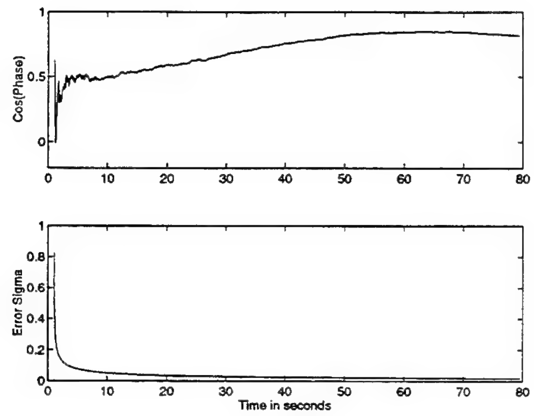


FIGURE 5.6 CosΦ_P Estimate from Kalman Filter

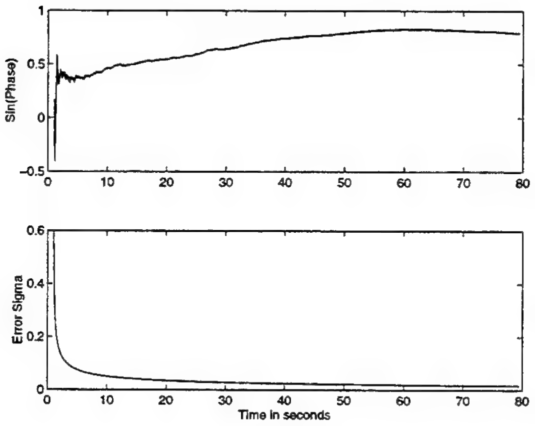


FIGURE 5.7 SinΦ_P Estimate from Kalman Filter

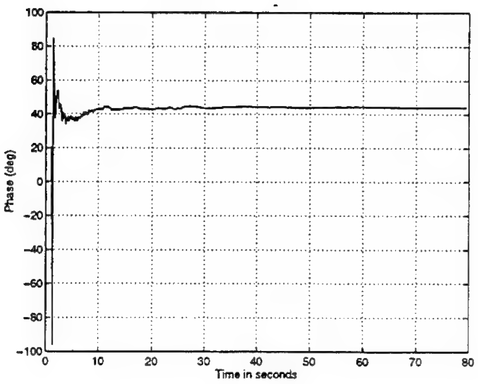


FIGURE 5.8 Phase Estimate from ArcTan of States 4 and 5

5.2.2.2 Performance in Noise and Lateral Rate Gyro Errors

Similar to the fast spinning LLSE algorithm of Section 3.2, the roll estimation process is not sensitive to the accuracy of the y and z body axis rate gyro measurement accuracies. As long as the relative amplitudes of the lateral gyros are significantly different, the phase angle estimation uses the rate of change of amplitude of each signal, not the signal value itself. This technique allows for a very robust routine in terms of lateral rate gyro errors. Figure 5.9 shows the simulation results with all the y and z rate gyro errors active at once (i.e. bias, g sensitivity, noise on all three gyros, scale factor and non-orthogonality). No significant degradation in performance results.

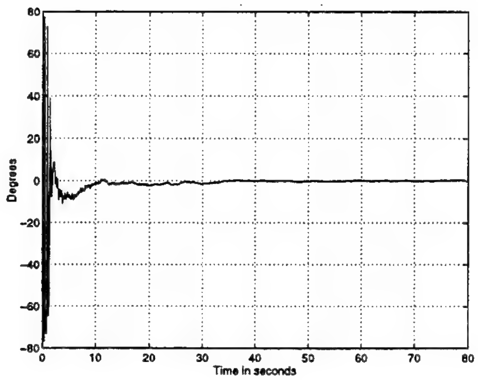


FIGURE 5.9 Roll Error with Y and Z Body Axis Rate Gyro Errors All Active

5.2.2.3 Performance in X Axis Rate Gyro Errors

Figure 5.10 shows that x axis rate gyro bias, g sensitivity, and noise on all three gyros have little affect on algorithm performance.

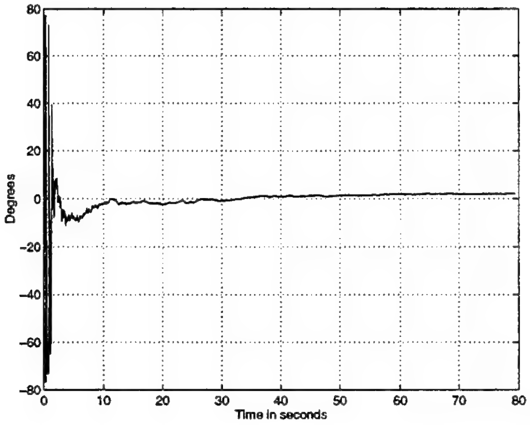


FIGURE 5.10 Roll Error with X Axis Bias, G Sensitivity, and Noise

When the simulation is rerun with x axis scale factor error only, the roll estimation performance is degraded as shown in Figure 5.11 . Note the slope of the roll error curve is exactly 1.44 deg/sec or the amount of error introduced into "P" (the shell roll rate estimate) by the x axis rate gyro 2000ppm scale factor error at 2Hz. The filter is very sensitive to the estimated value of the roll rate "P". Referring back to EQ 5.2.1.7 , recall that P is used in the matrix C(t) to relate both the sinΦ_P and cosΦ_P states to their respective measurements. The Kalman filter implementation integrates the roll angle estimate error introduced by x axis scale factor. This error could be substantially reduced by adding states to the navigation filter to estimate x axis scale factor error from accelerometer information and GPS as discussed briefly in Section 3.2.4

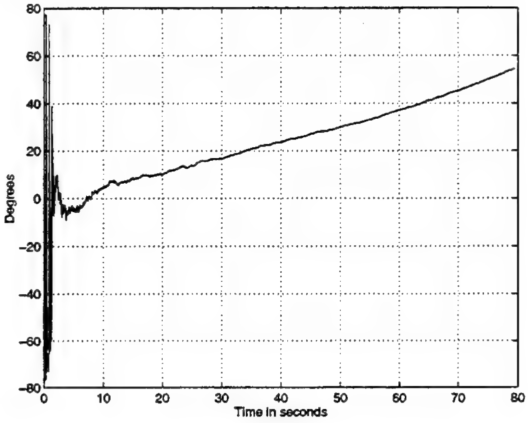
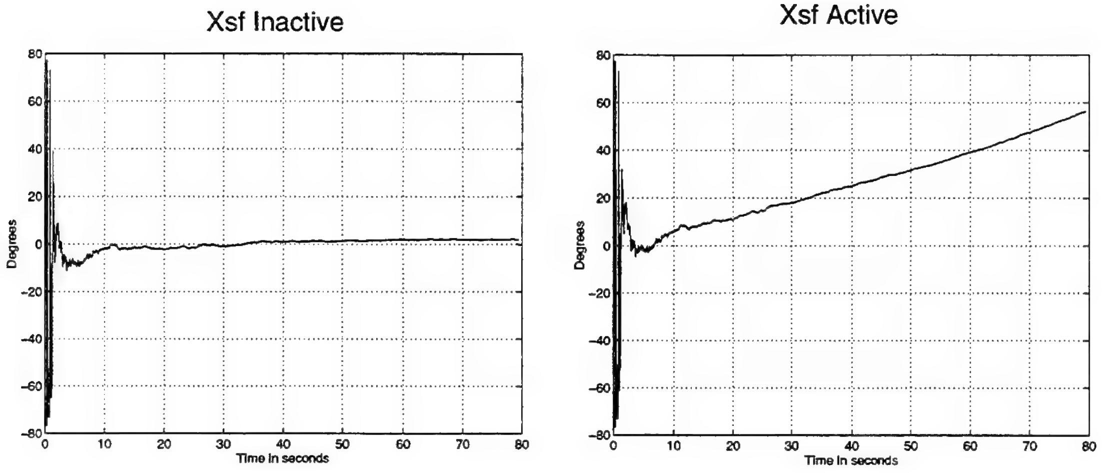


FIGURE 5.11 Roll Error with X Axis Scale Factor

5.2.2.4 Performance Summary for Fast Spinning Roll Estimation Kalman Filter

Total system performance was demonstrated by first simulating the shell at a spin rate of 2Hz with all the errors active, and re-running the simulation with all errors except the x axis rate gyro scale factor. The Kalman Filter developed in this section provides roll error accuracies with a standard deviation of 0.8777 deg in the case where x axis scale factor error is not included (Figure 5.12). This is in comparison to a standard deviation of 4 degrees for the LLSE fast spinning algorithm under the same conditions. If x axis scale factor is not removed, the resulting roll error performance is also shown in Figure 5.12 and results in prohibitive roll estimation errors.


Xsf Inactive

Xsf Active

FIGURE 5.12 Complete Roll Error Performance for 2Hz Kalman Filter with and without X Axis Scale Factor Error

5.3 Kalman Filter Implementation of Slow Spinning Algorithm

5.3.1 State Space Realization for Algorithm

The equations developed for the slow spinning shell algorithm in Section 4.2 are repeated here for the development of the slow spinning Kalman filter. In the slow spinning case, cosϕ and sinϕ can be estimated directly from the lateral gyro data.

$$\omega_y^b = Q\cos\phi + R\sin\phi \qquad \text{EQ (5.3.1.1)}$$

$$\omega_z^b = R\cos\phi - Q\sin\phi \qquad \text{EQ (5.3.1.2)}$$

A three state Kalman filter is developed using the rate gyro signals about three axis as observations of the three state realization where the first state is the shell roll rate "P", the second is cosϕ, and the third is sinϕ (EQ 5.3.1.3). The lumped lateral rate gyro biases are not included as states because they are inconsequential at small roll rates.

$$\begin{bmatrix} \omega_x^{\ b} \\ \omega_y^{\ b} \\ \omega_z^{\ b} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Q & R \\ 0 & R & -Q \end{bmatrix} \cdot \begin{bmatrix} P \\ \cos\phi \\ \sin\phi \end{bmatrix} + \underline{v} \qquad \text{EQ (5.3.1.3)}$$

where...

$\omega_x^{\ b}$ is the x body axis rate gyro output

$\omega_y^{\ b}$ is the y body axis rate gyro output

$\omega_z^{\ b}$ is the z body axis rate gyro output

ϕ is the shell roll angle at the current time

P is the shell roll rate estimate

Q is the average shell pitch over rate determined priori

R is the average shell yaw rate determined priori

This is of the form...

$$\underline{y}(t) = C(t) \cdot \underline{x}(t) + \underline{v} \qquad \text{EQ (5.3.1.4)}$$

The state vector is time varying because of the small roll rate P. This information is incorporated into the prediction portion of the state space realization as developed below.

$$\cos(\phi(t+\Delta t)) = \cos(\phi(t) + P\Delta t) \qquad \text{EQ (5.3.1.5)}$$

$$\sin(\phi(t+\Delta t)) = \sin(\phi(t) + P\Delta t) \qquad \text{EQ (5.3.1.6)}$$

$$\cos\phi(t+\Delta t) = \cos\phi(t) \cdot \cos(P\Delta t) - \sin\phi(t) \cdot \sin(P\Delta t) \qquad \text{EQ (5.3.1.7)}$$

$$\sin\phi(t+\Delta t) = \sin\phi(t) \cdot \cos(P\Delta t) + \cos\phi(t) \cdot \sin(P\Delta t) \qquad \text{EQ (5.3.1.8)}$$

which becomes...

$$\underline{x}(t+1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(P\Delta t) & -\sin(P\Delta t) \\ 0 & \sin(P\Delta t) & \cos(P\Delta t) \end{bmatrix} \cdot \underline{x}(t) + \underline{w} \qquad \text{EQ (5.3.1.9)}$$

where P is the shell roll rate estimate, Δt is the discrete time interval and EQ 5.3.1.9 is of the form...

$$\underline{x}(t+1) = A(t)\cdot\underline{x}(t) + \underline{w} \qquad \text{EQ (5.3.1.10)}$$

This state space realization is used in combination with the discrete Kalman Filtering equations to provide an recursive roll angle estimate every discrete time step by taking the arctangent of states 2 and 3 outside of the Kalman Filter.

$$\hat{\phi} = \operatorname{atan}\left(\frac{\sin\phi}{\cos\phi}\right) \qquad \text{EQ (5.3.1.11)}$$

5.3.1.1 Error Modelling of Observation Noise

The observation covariance matrix "$\mathcal{R}$" for the slow spinning case is simply the covariance of the three axis micro rate gyro outputs. These measurements have uncorrelated noise in their outputs with variance as already derived in EQ 5.2.1.1 .

$$\underline{y} = \begin{bmatrix} \omega_x^{\ b} \\ \omega_y^{\ b} \\ \omega_z^{\ b} \end{bmatrix} \qquad \text{EQ (5.3.1.1.1)}$$

$$\mathcal{R} = E\left[\underline{y}^T\cdot\underline{y}\right] \qquad \text{EQ (5.3.1.1.2)}$$

$$\mathcal{R} = \begin{bmatrix} 0.007162 & 0 & 0 \\ 0 & 0.007162 & 0 \\ 0 & 0 & 0.007162 \end{bmatrix} \qquad \text{EQ (5.3.1.1.3)}$$

5.3.1.2 Initial State and Initial Error Covariance Matrix

The filter is initialized with an initial estimate of the state vector $\overline{x}_0$ and the error covariance of the initial state estimate $\mathcal{P}_0$. The initial state estimate of the roll rate is simply the pure x axis rate gyro measurement at t_0 and the corresponding error covariance of this estimate is Var($\omega_x{}^b$). Guessing the roll angle to be 0 degrees provides and initial estimate of cosine and sine of roll to be 1 and 0 respectively. The error covariance of these initial estimate is the same as previously derived in Section 5.2.1.2 on page 76.

$$\bar{x}_0 = \begin{bmatrix} \omega_x{}^b(t_0) \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} P(t_0) \\ \cos\phi_0 \\ \sin\phi_0 \end{bmatrix} \qquad \text{EQ (5.3.1.2.1)}$$

$$\mathcal{P}_0 = \begin{bmatrix} 0.007162 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \qquad \text{EQ (5.3.1.2.2)}$$

5.3.2 Kalman Filter Performance

This Kalman Filter was coded into the Ada simulation for Monte Carlo analysis. The trajectory modelled is the same 30Km 155mm howitzer simulation described in Section 6.3 with a 0.04Hz shell roll rate. GPS measurements were not used for any of these simulations. The shell pitch over rate “Q” was set to a constant value of -1 deg/sec and the shell yaw rate “R” was set to a constant value of -0.01 deg/sec. The Kalman Filter performance in relation to the modelled micro gyro errors specified in Section 6.1 is presented below.

5.3.2.1 Performance in Noise Only

Figure 5.13 presents the roll error from the slow spinning Kalman Filter with micro gyro noise the only error source active.

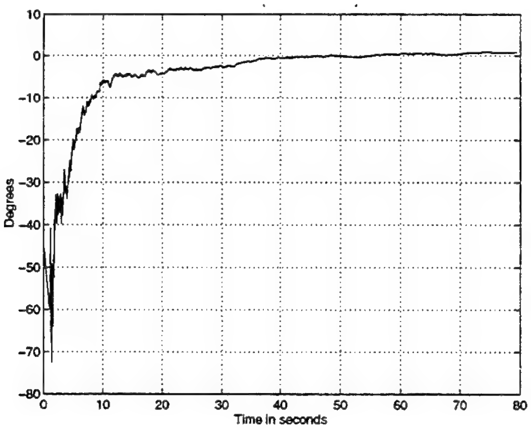


FIGURE 5.13 Kalman Filter Performance in Noise at 0.04Hz

The performance of the individual states in the Kalman Filter is shown in Figure 5.14 , Figure 5.15 , and Figure 5.16 . Notice that the estimates of sinϕ(t) and cosϕ(t) grow outside their expected range of between plus and minus one. They could have been constrained in an non-linear manner so that the sum of the squares is held equal to one.

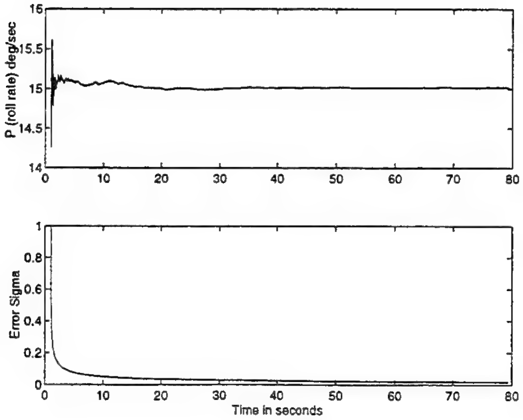


FIGURE 5.14 Roll Rate "P" Estimate From Kalman Filter

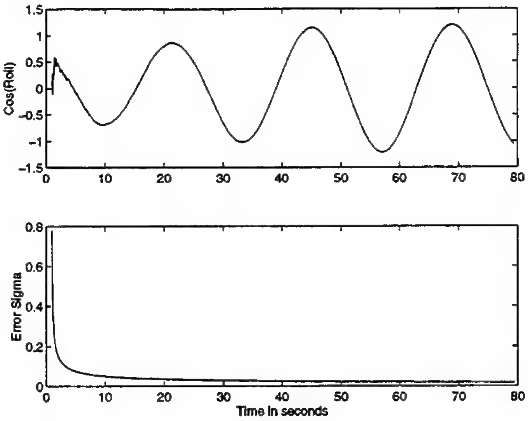


FIGURE 5.15 Cosϕ Estimate from Kalman Filter

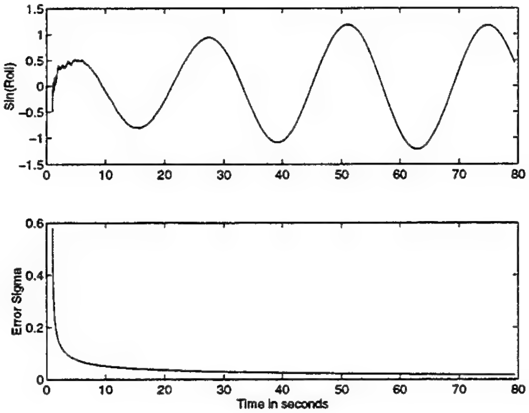


FIGURE 5.16 Sinϕ Estimate from Kalman Filter

5.3.2.2 Performance with X, Y, and Z Axis Rate Gyro Bias Active

Figure 5.17 shows the simulated filter roll error performance when bias errors are activated on all three axis rate gyros as well as gyro noise.

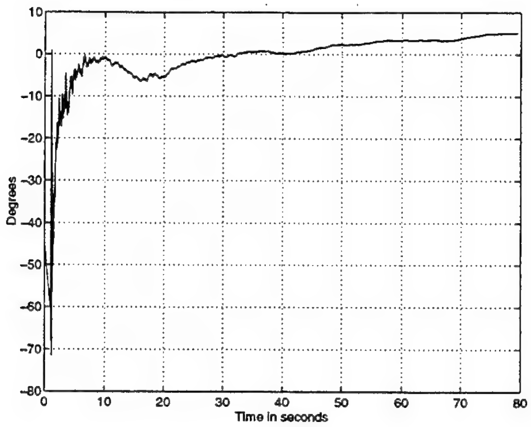


FIGURE 5.17 Roll Error Performance in Noise with X, Y, and Z Axis Rate Bias Active

5.3.2.3 Performance with X, Y, and Z Axis Rate Gyro Scale Factor Active

Figure 5.18 shows the simulated filter roll error performance when scale factor errors are activated on all three axis rate gyros as well as gyro noise.

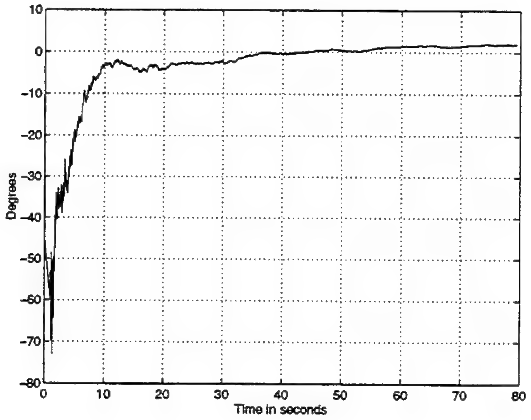


FIGURE 5.18 Roll Error Performance in Noise with X, Y, and Z Scale Factor Errors Active

5.3.2.4 Performance with X, Y, and Z Axis Rate Gyro G-Sensitivity Active

Figure 5.19 shows the simulated filter roll error performance when g-sensitivity errors are activated on all three axis rate gyros as well as gyro noise.

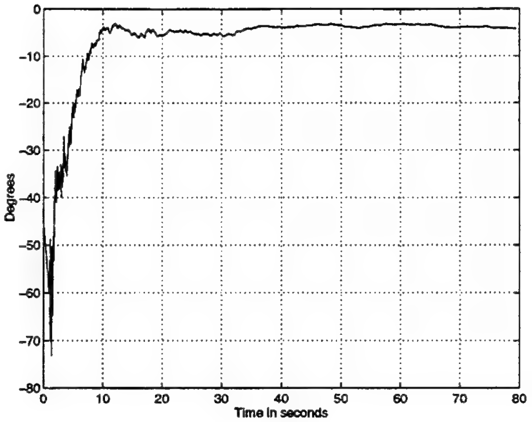


FIGURE 5.19 Roll Error Performance in Noise with X, Y, and Z Axis G Sensitivity Active

Figure 5.20 shows the resulting performance with only the x axis g sensitivity active. The deceleration due to the aerodynamic drag causes a large g-load along the x axis which decreases as the shell decelerates throughout the trajectory and results in the degradation in filter performance. The x axis g sensitivity error results in a bias error of -4.4859 degrees.

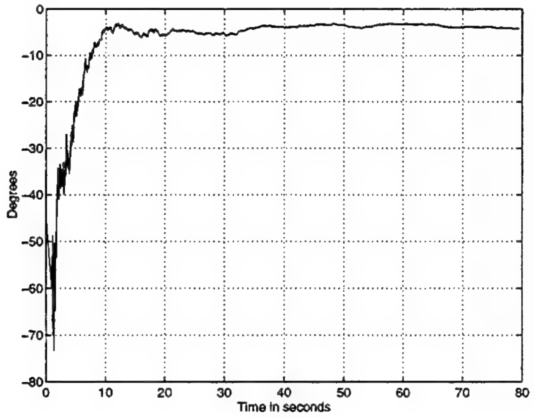


FIGURE 5.20 Roll Error Performance in Noise with only the X Axis G Sensitivity Active

5.3.2.5 Non-Orthogonalities and Misalignment Errors

The simulated trajectory was run with y and z axis non-orthogonality active to demonstrate the effects of physical misalignment of the gyros. The y axis was misaligned 96 mrad (5.5 degrees) in the yz plane and the z axis was likewise misaligned 96 mrad in the yz plane. Figure 5.21 shows the resulting roll error performance. The large errors introduced by the simulated misalignment emphasize the need for precision calibration upon MIMU assembly and installation in the shell to reduce misalignment type errors to much less than 96 mrad.

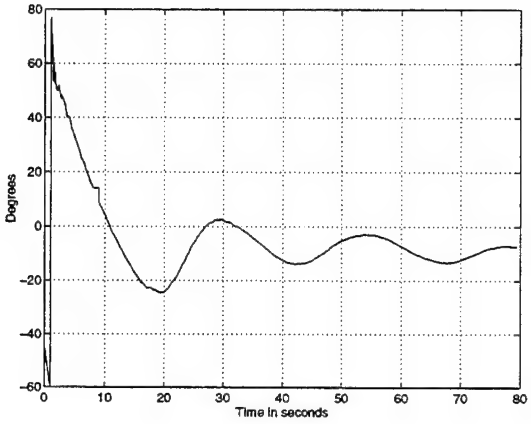


FIGURE 5.21 Roll Error Performance in Noise with Y and Z Axis Rate Gyro Non-Orthogonality of 96 mrad

5.3.2.6 Performance Summary for Slow Spinning Roll Estimation Kalman Filter

Total system performance was demonstrated by simulating the shell at a spin rate of 0.04Hz with all the errors active (except non-orthogonality). The Kalman Filter developed in this section provides roll error accuracies with a standard deviation of 2.1962 deg and a mean of -1.8227 deg in this case (Figure 5.22). This is in comparison to a standard deviation of 8.4252 degrees and a mean of -2.248 degrees for the LLSE slow spinning algorithm under the same conditions.

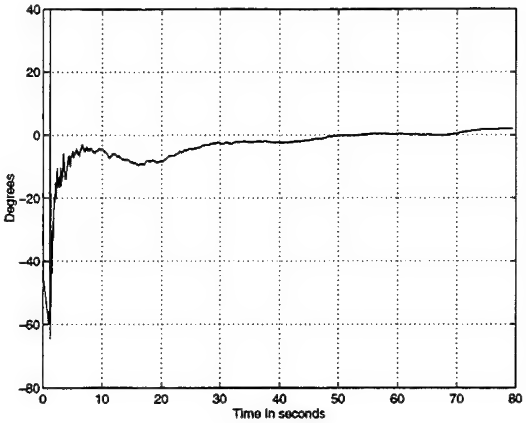


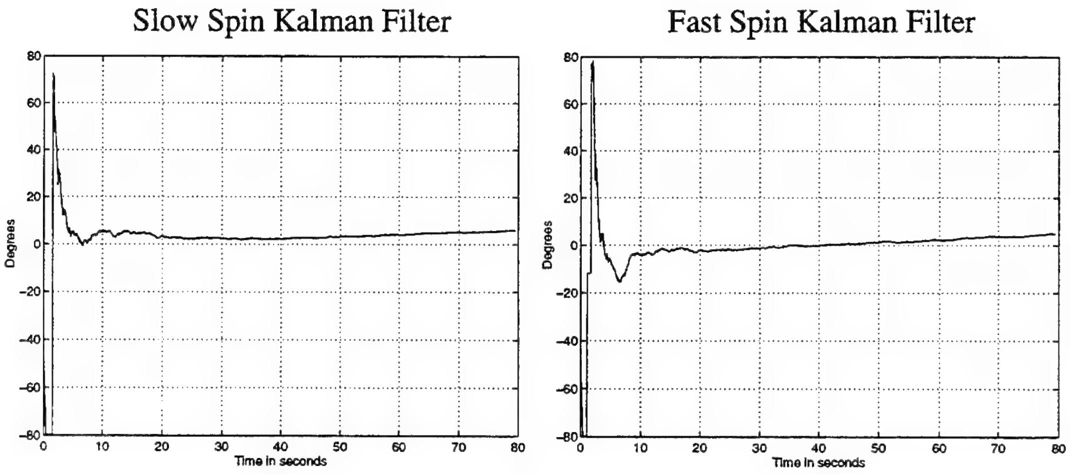
FIGURE 5.22 Roll Error Performance of Slow Spinning Kalman Filter with all Errors Active

5.4 Performance Crossover Point Between the Fast and Slow Spinning Kalman Filters

The analysis of Section 4.4 on page 62 is repeated for the two Kalman filters by simulating both routines (with all micro gyro errors active) at different roll rates and comparing the resulting roll error performance.

5.4.1 Comparing Filter Performance at Small Roll Rates

The slow spinning Kalman filter was much more robust in frequency than the slow spinning LLSE algorithm of Section 4.2. The performance crossover point is difficult to quantify exactly by simply observing the roll error performance, but crude comparisons yield very similar roll error performance for both filters at about 0.25Hz shell roll rate as shown in Figure 5.23 .


Slow Spin Kalman Filter

Fast Spin Kalman Filter

FIGURE 5.23 Roll Error Performance for both Kalman Filters at 0.25Hz Roll Rate

5.4.2 Performance when the Shell Spin Rate is 0Hz

Roll error performance gets worse when the shell is not spinning at all. This is no surprise for the fast spinning Kalman filter of Section 5.2 because the existence of a shell spin rate was fundamental to the development of the state space realization for the fast spinning case. The roll error performance of the fast spinning Kalman filter when the shell spin rate is 0Hz and all micro gyro errors are active is shown in Figure 5.24 .

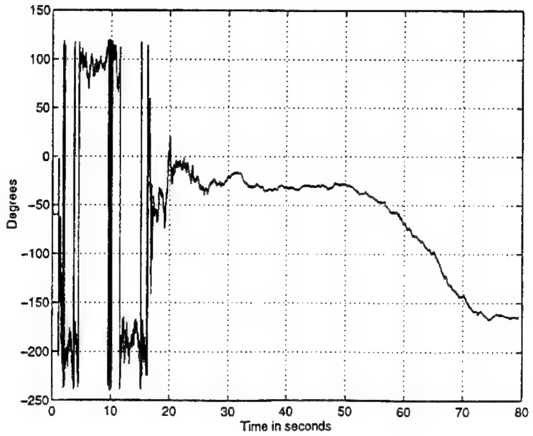


FIGURE 5.24 Fast Spinning Kalman Filter Roll Error for Non Spinning Case (0Hz)

The slow spinning Kalman filter state space realization was derived from the slow spinning LLSE algorithm of Section 4.2 on page 56 which began by solving the non spinning shell case and then correcting the errors incurred when the shell spin rate is small. However, the performance of the slow spinning Kalman filter in the non spinning shell case as shown below in Figure 25 is much worse than the roll error performance under the same conditions for a slowly spinning case of 0.04Hz from Figure 5.22 .

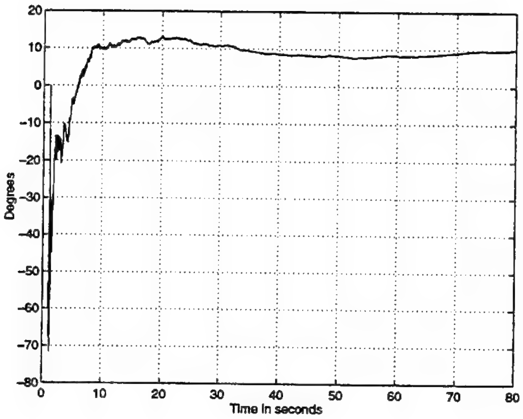


FIGURE 5.25 Slow Spinning Kalman Filter Roll Error for Non Spinning Case (0Hz)

What is the cause of the slow spinning Kalman filter performance degradation in the non spinning case? The simulation was run several times isolating the different micro gyro errors to observe their individual impact. All these runs were similar to the slowly spinning errors discussed in Section 5.3 except for the y and z axis (lateral rate gyros) bias error shown below in Figure 5.26 .

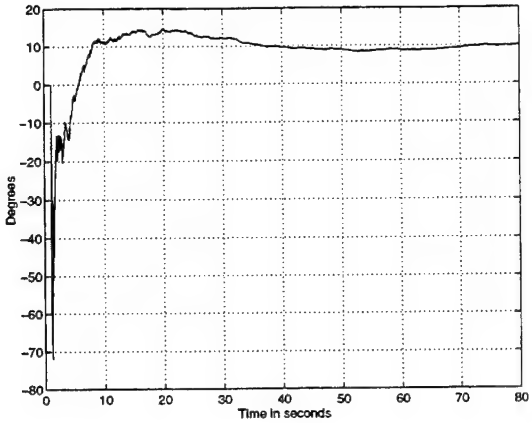
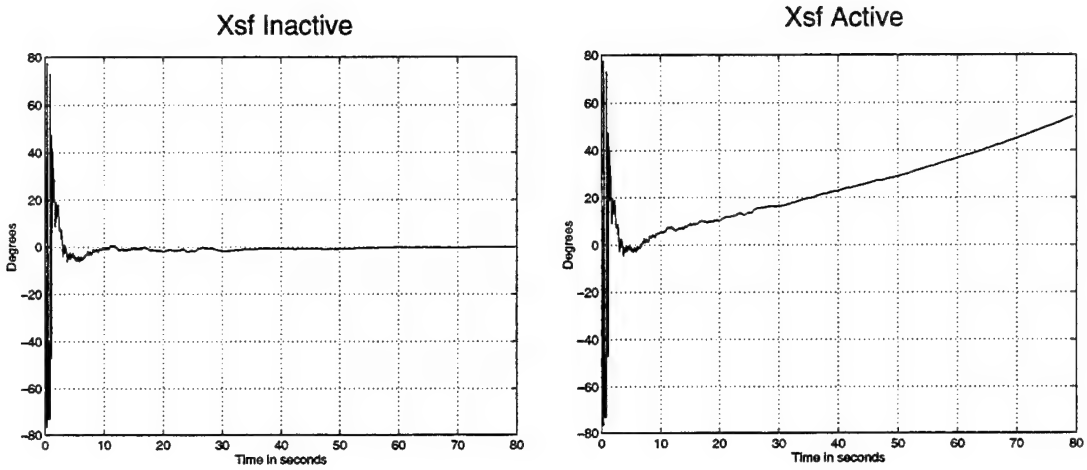


FIGURE 5.26 Slow Spinning Kalman Filter Roll Error with Y and Z axis Rate Gyro Bias Active and Noise with a Shell Roll Rate of 0Hz

The state space realization developed in Section 5.3.1 uses the y and z axis rate gyro signals as direct observations of sine and cosine of the roll angle. When these signals are biased, the resulting estimates of sine and cosine of the roll angle are biased as well. The arctangent of these biased estimates results in the 10 degree roll error bias shown in Figure 5.26 . When the shell is spinning slowly, the errors are modulated by the roll motion and tend to cancel, which explains the improved performance at 0.04Hz. In conclusion, a small roll rate near 0.04Hz produces the best performance for either Kalman filter.

5.4.3 Performance of the Slow Spinning Kalman Filter at 2Hz

The slow spinning LLSE batch processing algorithm produced useless estimates at roll rates greater than 0.15Hz because the formulation of the batch processing technique did not adequately account for roll rate dynamics. However, the slow spinning Kalman filter does account for these dynamics in the prediction step by including the shell roll rate estimate in the transition matrix A(t) to form the state space model. Given this, the results shown in Figure 5.27 for the slow spinning Kalman filter performance at the "fast" 2Hz shell spin rate are not surprising.


Xsf Inactive

Xsf Active

FIGURE 5.27 Slow Spinning Kalman Filter Performance with Shell Roll Rate of 2Hz

In conclusion, both the fast and slow spinning Kalman filters have similar performance characteristics at roll rates from 0.25Hz to 2Hz. When the shell is not spinning, the slow spinning Kalman filter does a much better job of finding the fixed roll angle than the fast spinning Kalman filter.

Chapter 6

Computer Simulation Development

6.1 Simulation Overview

A high fidelity Monte Carlo simulation of a generic artillery shell equipped with an MIMU and a 24 channel GPS receiver was developed by modifying the Embedded GPS Doppler Test Facility software model of the UH-60 Blackhawk Helicopter documented in [10] . The simulation is written in Ada to run on a Sun Microsystems workstation. The modifications to the existing GPS/Doppler simulation are summarized as follows...

- Rename and modify variables to define specific artillery shell characteristics
- Eliminate unnecessary sensor models including doppler navigation models, gyro magnetic compass models, barometric altimeter models, and vertical gyro models
- Add MIMU software model of micro gyro and micro accelerometer triads with appropriate errors
- Adapt GPS model to shell platform by updating antenna installation and gain objects
- Add state estimation objects for performance analysis of roll estimation algorithms developed for this thesis
- Add variables to the timeliner script to specify artillery shell physical characteristics, sensor error characteristics, algorithm performance variables, and sensor installation locations

The simulation architecture is depicted in Figure 6.1 .

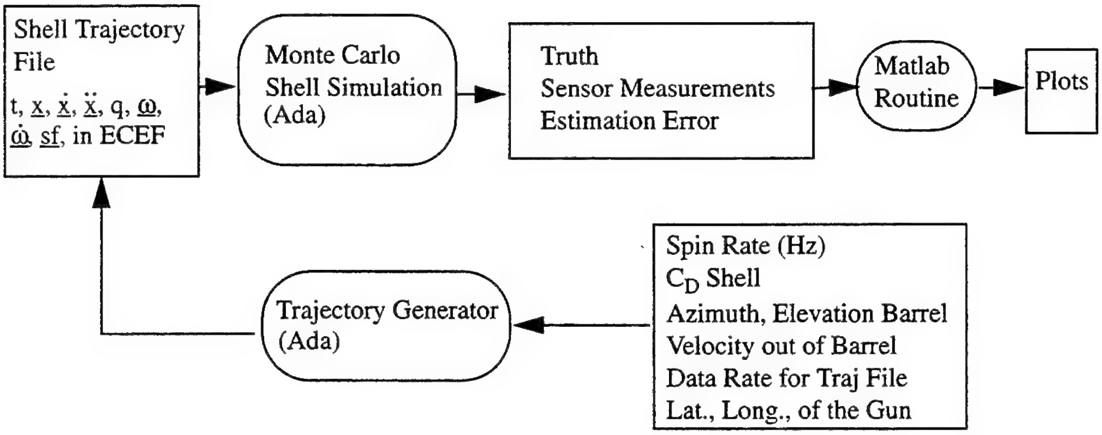

Shell Trajectory File
t, $\underline{x}$, $\dot{\underline{x}}$, $\ddot{\underline{x}}$, q, $\underline{\omega}$, $\dot{\underline{\omega}}$, $\underline{sf}$, in ECEF
Monte Carlo Shell Simulation (Ada)
Truth
Sensor Measurements
Estimation Error
Matlab Routine
Plots
Spin Rate (Hz)
C_D Shell
Azimuth, Elevation Barrel
Velocity out of Barrel
Data Rate for Traj File
Lat., Long., of the Gun
Trajectory Generator (Ada)

FIGURE 6.1 Monte Carlo Simulation Overview

The inertial Ada Artillery Shell Simulation reads data from a trajectory file in Earth Centered Earth Fixed (ECEF) coordinates. This data consists of time, position, velocity, acceleration, specific force, attitude quaternion, angular rate, and angular acceleration. This trajectory file is created by another Ada simulation described in Section 6.3. Operating at a specified system rate (typically 1000Hz), the Artillery Shell Simulation reads in the truth data for the shell center of mass, determines the truth data for each sensor location, exercises each sensor model with the error characteristics specified in the timeliner script, exercises state estimation objects which use the modelled sensor data, and finally collects and reports the truth and measured parameters to an output file. The output file is reduced via a Matlab routine for observation of the system performance.

6.2 Navigation Sensor Performance

6.2.1 Micro Mechanical Rate Gyro Errors

Table 1 records the current state of technology for Draper micro gyro accuracy [3] .

TABLE 1. Micro Gyro Specifications

Stability (deg/hr)	480.0
Scale Fact (ppm)	2000.0
G-Sens (deg/hr/g)	720.0
Misalign (mrad)	96.0
Rand Walk (deg/rt-hr)	9.2

This section will describe the error mechanisms present in the micro gyros. This information is not clearly documented in any single source, however many of the terms are described in reference [7] .

- *Stability Error*

Sometimes denoted as "drift", rate gyro angular output wanders away from truth at the RMS rate denoted as "stability error" in Table 1 . Since the time constant is large (between 1 hour and 10 hours) for stability error, this effect has been modelled as a constant bias for the artillery shell simulation of an 80 second trajectory.

- *Misalignment Error*

The job of the navigation system is to determine the angular attitude of the vehicle in inertial space. The sensors can only detect the angular orientation of the sensor. Misalignment error deals with the imprecision in mounting the assembled MIMU unit onto the vehicle. Misalignment is the difference between the actual orientation of the equipments sensitive axis and the intended orientation of that axis. This is a constant angular error and is normally kept small by precision mounting and measuring techniques during installation.

- *Non-Orthogonality Error*

Non-orthogonality error refers to imprecision in assembling the MIMU unit itself. The MIMU consists of three micro gyros and the intent is to mount them perfectly along 3 orthogonal axes. Non-orthogonality error results when one gyro input axis leans into the plane containing the remaining two gyro input axes. This non-orthogonal gyro will detect components of the angular rates about the other axes. Figure 6.2 shows the relationship between the unit vectors in the Navigation Body (NB) frame and a unit vector corresponding to the misaligned z axis gyro. The error due to non-orthogonality in the z axis is found by determining the components of each NB frame angular rate sensed by the misaligned rate gyro. This relationship is given in EQ 6.2.1 .

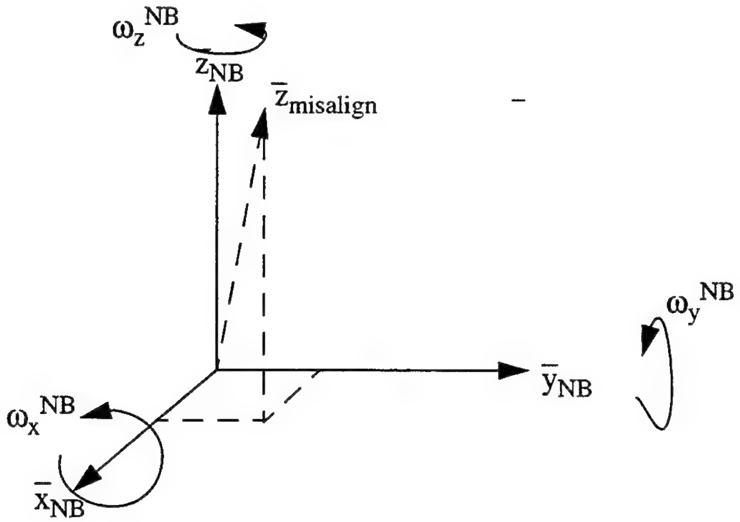


FIGURE 6.2 Non-Orthogonality Error Depiction

$$E_z = (\bar{z}_{NB} \cdot \bar{z}_{mis} - 1) \cdot \omega_z^{NB} + (\bar{x}_{NB} \cdot \bar{z}_{mis}) \cdot \omega_x^{NB} + (\bar{y}_{NB} \cdot \bar{z}_{mis}) \cdot \omega_y^{NB} \quad \text{EQ (6.2.1)}$$

- *Scale Factor Error*

The micro gyro scale factor error is calculated as a percentage of the true angular rate sensed by the rate gyro. Scale factors produce an error in the measured angular rate of a magnitude that is proportional to the true angular rate being measured. Scale factor errors are most troublesome at high angular rates.

$$Error_{scalefactor} = ScaleFactor \cdot \omega_{sensed} \qquad \text{EQ (6.2.2)}$$

- G Sensitivity Error

“Sensitivity is a change in output to a change in an undesirable or secondary input.”[7] Gyroscope output variation due to accelerations incident on the instrument are called g-sensitivity error. This produces rate bias error proportional to the amount of specific force in a maneuver and lasting the duration of the specific force. Note: specific force is equal to the inertial acceleration of a body minus gravitational acceleration.

$$Error_g = g_{sensitivity} \cdot sf_{sensed} \qquad \text{EQ (6.2.3)}$$

- *Angle Random Walk*

Micro gyroscopes are noisy sensors. Figure 6.3 is actual bench test data from Draper laboratories showing the noise level of the micro gyro. The micro gyro acts as an integrator of the sensed angular rate. The actual sensor output of $\Delta\theta$ integrates the noise shown in Figure 6.3 to produce a smoother signal that randomly wanders through a certain maximum error. Angle random walk is modelled in the Monte Carlo simulation by generating white noise at the system data rate and multiplying by the appropriate constant to result in the specified random walk characteristics.

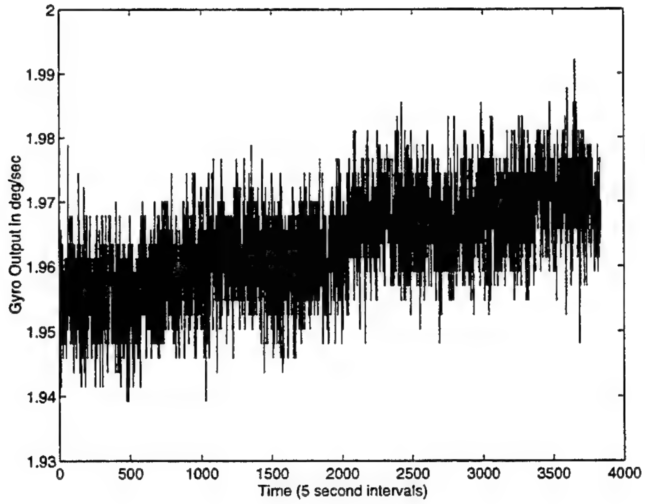


FIGURE 6.3 Gyro Output From Static Bench Test

- *Temperature Sensitivity*

The micro gyro is a good thermometer. Temperature bias of the micro gyro is also evident in the bench test data shown in Figure 6.4 . Temperature bias will have to be removed by the navigation algorithm to preclude large errors in the integrated solution.

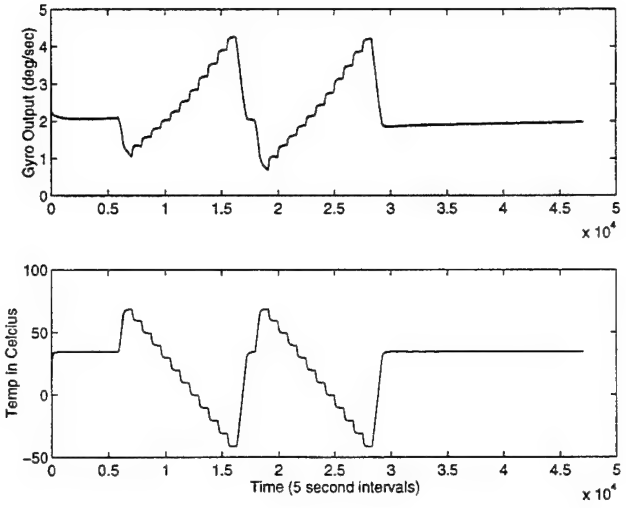


FIGURE 6.4 Gyro Output vs. Temperature from Bench Test

6.2.2 Micro Mechanical Accelerometer Errors

Table 1 records the current state of technology for Draper micro accelerometer accuracy [3] .

TABLE 2. Micro Accelerometer Specifications

Stability (millig)	58.0
Scale Fact (ppm)	4000.0
G^2-Sens (microg/g^2)	8000.0
Misalign (mrad)	11.0
Velocity Random Walk (cm/s/rt-hr)	340.0

The error terms have similar definitions whether they represent micro accelerometer errors or micro gyro errors. This section will describe the error mechanisms present in the micro accelerometers not explained earlier for the micro gyros.

- G^2 Sensitivity Error

Accelerometer output variation due to the square of the acceleration incident on the instrument's sensitive axis is called g^2 sensitivity error. This produces acceleration bias error proportional to the amount of specific force squared in a maneuver which lasts the duration of the specific force.

$$Error_{g^2} = g^2_{sensitivity} \cdot sf^2_{sensed} \qquad \text{EQ (6.2.1)}$$

- *Velocity Random Walk*

Micro accelerometers are noisy sensors. The micro accelerometer acts as an integrator of the measured specific force. The actual sensor output of ΔV measurements integrates the noise to produce a velocity signal that randomly wanders through a certain maximum error. Velocity random walk is modelled in the Monte Carlo simulation by generating white noise at the system data rate and multiplying by the appropriate constant to result in the specified random walk characteristics.

6.2.3 GPS Velocity Measurements

The Guided Munition Demonstration Program autopilot navigation filter will be restricted to use only the position and velocity measurements from the embedded GPS receiver's own navigation filter. The follow-on program will incorporate psuedorange and delta range measurements directly from the GPS receiver into a single navigation filter for a more optimal navigation solution. The Ada GPS simulation only models GPS psuedorange and delta range from each satellite, not position and velocity for a specific GPS receiver. Because GPS velocity measurements were required for the analysis in Chapter 2 , GPS velocity was crudely modelled by adding velocity noise ($\sigma = 0.1$m/s) to the true velocity from the trajectory file.

6.3 Trajectory Generation

The trajectory generating software was written in Ada to create a data file of variables that described a ballistic artillery shell trajectory in ECEF coordinates.

6.3.1 Artillery Shell Dynamics Modelling

The artillery shell was modelled as a point mass with a drag coefficient C_{D0}. The position and velocity of the shell were determined by integrating Newton's law of motion for a point mass. The point mass is influenced by the force of gravity, a drag force acting directly opposite the shell velocity vector, the Coriolis force due to the shell velocity in the rotating ECEF reference frame, and the rotational force due to the rotation of the earth. [6] A spinning shell will also experience an additional yawing moment due to the interaction of the spinning shell boundary layer with the atmosphere [5] . This effect is called the yaw of repose and is ignored by the trajectory generator.

$$F_{shell} = mg + C_{D0}\rho V_{rel}\left(\frac{Area}{2}\right) + 2 \cdot (\bar{\varpi}_E \otimes \bar{V}_{rel}) + \bar{\varpi}_E \otimes (\bar{\varpi}_E \otimes \bar{P}_{rel}) \quad \text{EQ (6.3.1)}$$

where...

$m = mass$
$C_{D0} = Drag\ Coefficient$
$\overline{V}_{rel} = Shell\ Velocity\ Vector\ Relative\ to\ Earth's\ Surface$
$\omega_E = Earth\ Rotation\ Rate$
$\overline{P}_{rel} = Shell\ Position\ Vector\ Relative\ to\ Earth's\ Surface$
$\rho = Density\ of\ Air$

The initial velocity V_0 is known and acts in the direction of the azimuth and elevation of the gun barrel. The initial position P_0 is the latitude and longitude of the gun's location. The program terminates when the position vector in the North East Down (NED) frame is positive in the "z" or "down" direction.

The trajectory generator assumes the shell nose or roll axis is perfectly aligned with the shell velocity vector at all times. The pitch and yaw attitudes are simply determined by calculating the angles defining the velocity vector in the NED frame. The angular rates in pitch and yaw are calculated by differentiating the change in attitude across the time step Δt. The shell roll rate and initial roll angle are constant numbers selected prior to program execution. The roll angle is derived by integrating the roll rate throughout the trajectory from the initial roll angle.

6.3.2 Artillery Shell Simulation Parameters

A single trajectory was modelled for all the analysis presented in this thesis. The shell roll rate along this single trajectory was varied between 0-2Hz to produce the variety of results discussed herein (note: the trajectory generator only considers constant shell roll rates). Table 3 below documents the parameters (taken from [3]) that define the trajectory simulated and recorded in the trajectory files used by the Ada artillery shell simulation.

TABLE 3. Trajectory Simulation Parameters Used for Analysis

Trajectory Parameter	Value Used
Shell Diameter	155 mm
C_{D0}	0.0033
Initial Velocity	830 m/sec
Initial Latitude	40 deg north
Initial Longitude	70 deg west
Gun Barrel Azimuth	0 deg (due north)
Gun Barrel Elevation	38.75 deg

The preceding parameters produce an artillery shell trajectory that travels 30 km down range in about 79 seconds. The following set of figures show the details of the simulated trajectory.

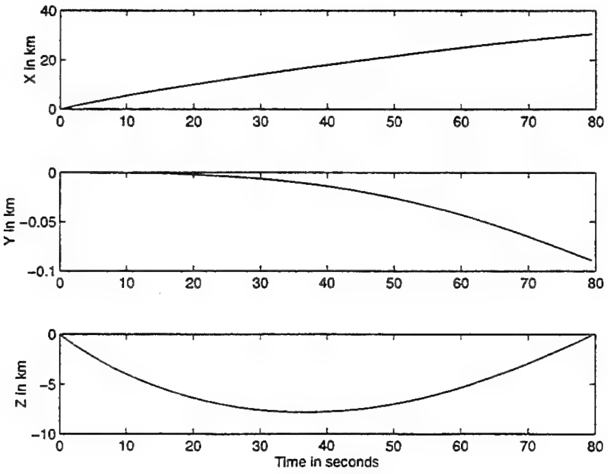


FIGURE 6.5 30km Simulated Trajectory True Position in NED

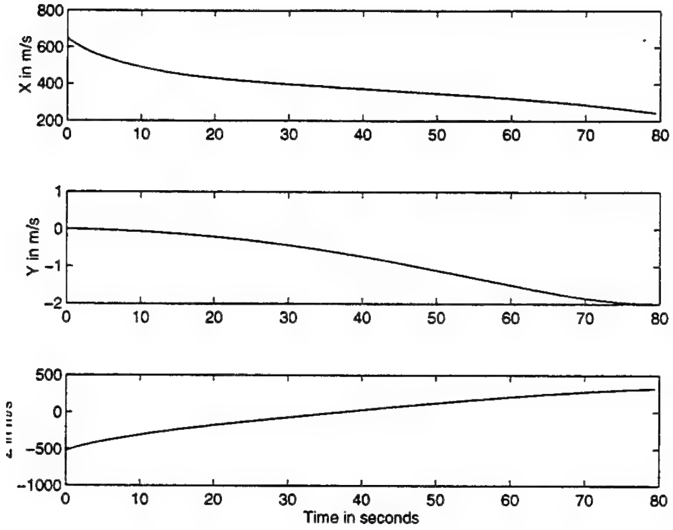


FIGURE 6.6 30km Simulated Trajectory True Velocity in NED

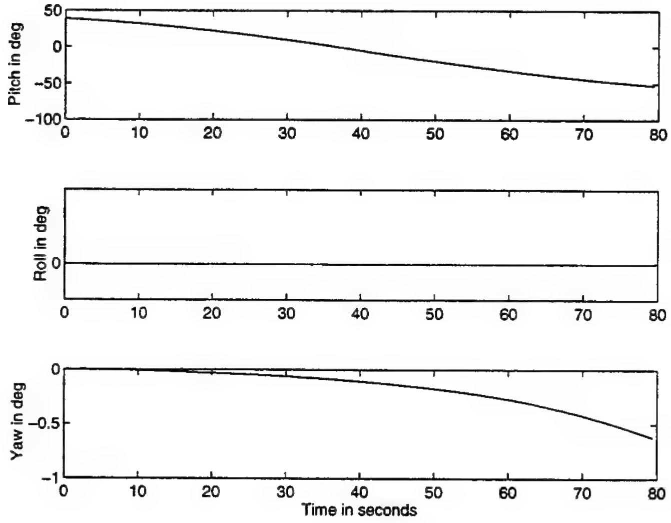


FIGURE 6.7 30km Simulated Trajectory True Attitude in Body Axis (Roll Rate 0Hz)

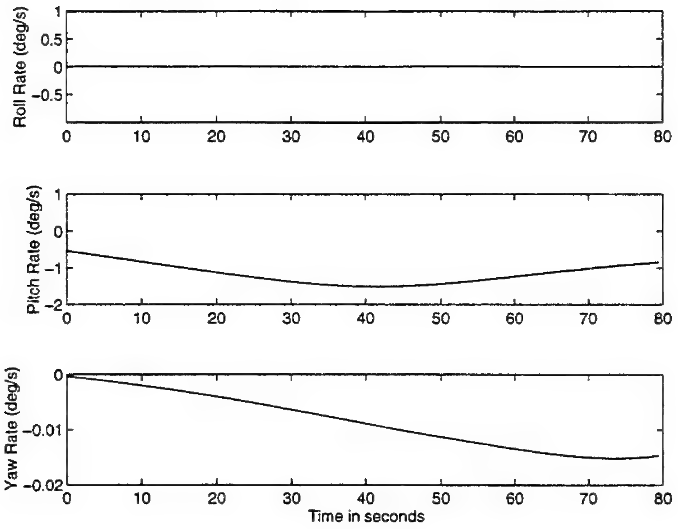


FIGURE 6.8 30km Simulated Trajectory True Attitude Rate in Body Axis (Roll Rate 0Hz)

Chapter 7

Conclusions and Future Research

7.1 Conclusions

The goal of this thesis was to develop a method of estimating the local vertical state of a ballistic artillery shell spinning at a rate from 0-2Hz. The shell was equipped with a GPS receiver providing position and velocity data at 1Hz and an MIMU providing $\Delta\theta$ and ΔV measurements at 1000Hz. In addition, any initial roll angle state information was considered lost due to the high dynamics (6200g and high shell spin rates) incurred when launching an artillery shell out of a cannon.

A method of estimating artillery shell pitch over rate (due to the gravity turn) and shell yaw rate (due to Coriolis force) from GPS velocity measurements was developed and its performance was simulated. This information was combined with data from the lateral micro rate gyros to produce two orthogonal signals from which the shell roll angle could be estimated. Several attempts to estimate the roll angle of a spinning shell were made and their simulated performance was documented. The most accurate algorithm was a deterministic linear least squares batch processing technique that estimated the roll angle within + or - 4 degrees at a shell spin rate of 2Hz throughout the trajectory (without x axis scale factor errors included). Each micro gyro error modelled was individually analyzed for performance degradation of the fast spinning algorithm. Misalignment errors and non-orthogonality errors had a large impact when a component of the roll rate was projected onto one or both of the lateral micro gyros. This created a large bias in the lateral rate gyro data that would ruin the algorithm performance if not estimated and removed. Errors in the spin axis rate gyro due to scale factor, misalignment, or non-orthogonality also contributed a significant performance degradation. An attempt to estimate x axis rate gyro error by Fourier transform of lateral rate gyro data was ineffective. In conclusion, it was noted that precision calibration and manufacturing of the micro gyro would be required to reduce non-orthogonality and misalignment errors. Also, x scale factor error would have to be estimated and removed in the system navigation filter using GPS information.

The spinning shell algorithm performance became degraded as the shell roll rate approached 0 Hz. Therefore, a slowly spinning roll estimation algorithm was developed using a modified deterministic linear least squares batch processing technique. The simulated performance of this algorithm provided a roll angle estimate within + or - 7 degrees throughout most of the shell trajectory at slow spin rates. The fast and slow spinning roll estimation algorithm performance crossover point was determined to be at a shell spin rate of about 0.08Hz.

The performance of both algorithms was unaffected when the shell pitch over rate and yaw rate estimates from GPS were replaced with constant numbers indicative of the average angular rates expected for shell pitch over and yaw. The algorithms did not rely on the value of the lateral rate gyro amplitudes but simply the change in amplitude observed across the roll period. Eliminating the need for GPS allowed the algorithms to be active during the 15 seconds required for GPS reacquisition after launch.

Two extended Kalman filter implementations (one for both the fast and slow spinning algorithms) were developed and each filter's performance was simulated. The fast spinning shell Kalman filter included five states and yielded roll estimates an order of magnitude better than the corresponding batch processing technique. Each micro gyro error was individually analyzed for performance impact on the filter. X axis scale factor error made the only significant performance degradation of all the errors modelled. A 2000ppm x scale factor error caused the roll estimate to linearly drift away from truth at a rate of 1.44 deg/sec.

The slow spinning shell Kalman filter included three states and also yielded roll estimates an order of magnitude better than the corresponding batch processing technique with all simulated errors active and the shell spin rate at 0.04Hz. Each micro gyro error was individually analyzed for performance impact on the filter. Non-orthogonality error also caused significant performance degradation when allowed into the slow spinning simulation and will have to be eliminated by sensor calibration. X axis G sensitivity error made the only significant performance degradation of all the errors modelled. X axis scale factor error was not significant at small roll rates. Also, when the shell trajectory was non spinning (0Hz), the performance was degraded. For the non spinning case, the lateral rate gyro bias errors could not cancel each other out and the resulting roll angle was biased from truth by 10 degrees (see Figure 5.26). Finally, the slow spinning Kalman filter performance at 2Hz was comparable to the fast spinning Kalman filter performance at that rate.

In summary, extended Kalman filter roll angle estimation of a spinning shell from inertial sensor data can provide accurate performance for shell roll rates between 0-2Hz. GPS is only required for correcting x axis scale factor error in the fast spinning shell case. Precision calibration and manufacturing of the MIMU will be required to reduce the effects of misalignment and non-orthogonality errors.

7.2 Future Research

Future studies should integrate the extended Kalman filter states from this thesis into the complete shell navigation filter and use the additional information available to estimate and remove x axis scale factor errors, g sensitivity errors, misalignment errors, and bias errors. Performance degradation due to noise in the shell roll rate and slowly varying roll rates should be studied as well as the effect of adding the yaw of repose dynamics to the trajectory behavior. Data quantization of the inertial micro sensors was not an error parameter in the Thesis simulation. An adequate hardware quantization level for the gyros needs to be determined. There may be a trade off between quantization level and update rate.

Appendix A

Roll Estimation For Low Noise Gyros

Solving EQ 3.2.1 and EQ 3.2.2 for sinϕ and cosϕ where ϕ is time varying produces the following relationships.

$$\cos\phi(t) = \frac{Q\omega_y^{\ b} + R\omega_z^{\ b}}{Q^2 + R^2} \qquad \text{EQ (A.1)}$$

$$\sin\phi(t) = \frac{R\omega_y^{\ b} - Q\omega_z^{\ b}}{Q^2 + R^2} \qquad \text{EQ (A.2)}$$

Notice also that...

$$\int\cos\phi d\phi = \sin\phi(t) = \int\frac{Q\omega_y^{\ b} + R\omega_z^{\ b}}{Q^2 + R^2}d\phi \qquad \text{EQ (A.3)}$$

$$\int\sin\phi d\phi = -\cos\phi(t) = \int\frac{R\omega_y^{\ b} - Q\omega_z^{\ b}}{Q^2 + R^2}d\phi \qquad \text{EQ (A.4)}$$

$$d\phi = Pdt \qquad \text{EQ (A.5)}$$

where P is the roll rate and is considered constant across the small interval dϕ. Integrating the noisy measurements will make extracting the roll information less suggestible to noise. The final algorithm uses estimates of yaw and pitch rate derived from the GPS Euler angle projection combined with the rate gyro data from the x, y and z body axes. This data is combined and integrated over time. The arctangent function provides the proper quadrant for the roll angle based on the value of sinϕ and cosϕ.

$$\tan\hat{\phi}(t) = \frac{\int_0^t \frac{\hat{Q}\omega_y^{\ b} + \hat{R}\omega_z^{\ b}}{\hat{Q}^2 + \hat{R}^2}\hat{P}dt}{\int_0^t \frac{\hat{Q}\omega_z^{\ b} - \hat{R}\omega_y^{\ b}}{\hat{Q}^2 + \hat{R}^2}\hat{P}dt} \qquad \text{EQ (A.6)}$$

$\hat{Q}$ = *Pitch Rate Estimate*
$\hat{R}$ = *Yaw Rate Estimate*
$\hat{P}$ = *Roll Rate Estimate* = $\omega_x^{\ b}$ = *Roll Axis Rate Gyro Output*
$\omega_y^{\ b}$ = *Pitch Axis Rate Gyro Output*
$\omega_z^{\ b}$ = *Yaw Axis Rate Gyro Output*

In the absence of sensor errors the algorithm performance is shown in Figure A.1 . The algorithm implementation waits 1 full rotation of the shell before beginning to estimate the roll angle. Since no initial conditions are available for the integration of the sine and cosine equations, the integrated sine and cosine data is biased and needs to be shifted to lie between 1 and -1. The algorithm watches each integral equation for 1 cycle to compute and remove the bias.

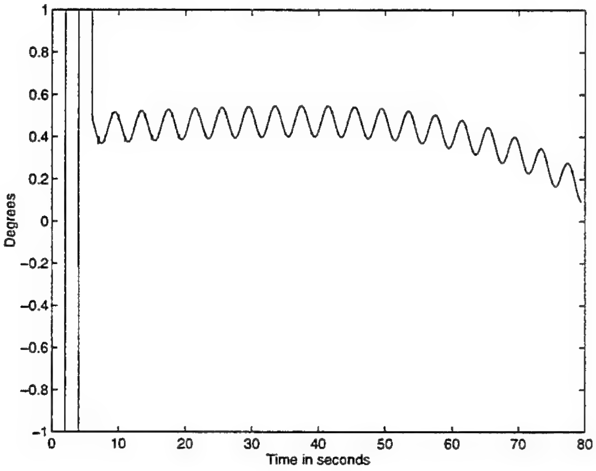


FIGURE A.1 Error in Estimated Roll Angle With Roll Rate = 1/4 Hz and Data Rate = 100Hz

This algorithm assumes we know the body axis angular rates $\omega_x{}^b$, $\omega_x{}^b$, and $\omega_x{}^b$. This information is provided by the micro gyros and is extremely noisy. Figure A.2 shows the simulated gyro output with noise for a trajectory with constant 1/4 Hz roll rate.

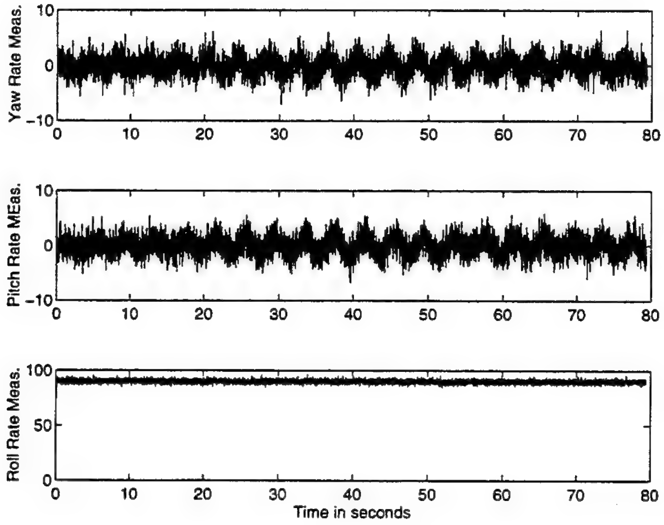


FIGURE A.2 Measured Angular Rates in Body Frame

Appendix B

Estimating Angular Rate from Noisy Gyro Data Using Non-Random Parameter Estimation

The observed gyro output “y” is modelled as the true angular rate “x” (a fixed deterministic parameter for the non-spinning case) plus zero mean white gausian noise “w” with some fixed variance (EQ B.1). A vector of observations of the gyro output corresponds

$$y = x + w \qquad \text{EQ (B.1)}$$

to several observations of the constant value “x” plus a vector of noise observations (EQ B.2). A maximum likelihood estimate (ML) of the non-random parameter “x” is

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ \dots \\ y_n \end{bmatrix} = x + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \dots \\ \dots \\ w_n \end{bmatrix} \qquad \text{EQ (B.2)}$$

obtained by finding the argument that maximizes the probability density function for **Y** given x [9] .

$$P_{Y|x} = \frac{1}{2\pi^{\frac{n}{2}} \cdot |\Lambda_Y|^{\frac{1}{2}}} \cdot e^{\frac{-1}{2} \cdot (Y - m_x)^T \cdot |\Lambda_Y|^{-1} \cdot (Y - m_x)} \qquad \text{EQ (B.3)}$$

where...

$$\Lambda_Y = \Lambda_w \qquad \text{EQ (B.4)}$$

$$m_x = x \qquad \text{EQ (B.5)}$$

so that...

$$P_{Y|x} = \frac{1}{2\pi^{\frac{n}{2}} \cdot |\Lambda_w|^{\frac{1}{2}}} \cdot e^{\frac{-1}{2} \cdot (Y-x)^T \cdot |\Lambda_w|^{-1} \cdot (Y-x)} \qquad \text{EQ (B.6)}$$

EQ B.6 is maximized when the exponent is equal to zero. This yields...

$$Y = \hat{x}_{ML} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ \dots \\ 1 \end{bmatrix} \qquad \text{EQ (B.7)}$$

Solving for the ML estimate yields the following relationship which means the best estimate of “x” is the average of the values observed in “y” [9] .

$$\hat{x}_{ML} = \frac{\sum_{i=1}^{N} y_i}{N} \qquad \text{EQ (B.8)}$$

If a vector has N observations of the shell spin rate, pitch rate or yaw rate in the body axis from a micro gyro, the estimated angular rates of the shell in the body axes are given by the following.

$$\hat{\omega}^b{}_x = \frac{\sum_{i=1}^{N} \omega^b{}_{xi}}{N} \qquad \text{EQ (B.9)}$$

$$\hat{\omega}^b{}_y = \frac{\sum_{i=1}^{N} \omega^b{}_{yi}}{N} \qquad \text{EQ (B.10)}$$

$$\hat{\omega}^b{}_z = \frac{\sum_{i=1}^{N} \omega^b{}_{zi}}{N} \qquad \text{EQ (B.11)}$$

Appendix C

Roll Estimation by Locating Signal Maximums and Minimums

Noise in the gyros completely corrupts the roll estimation algorithm presented in Appendix A . This is because the arctangent function is indeterminate when noisy gyro measurements are integrated and the resulting random walking signal cannot be easily constrained within + or - 1. This is illustrated in Figure C.1 which shows EQ C.1 and EQ C.2 for sinϕ and cosϕ for 1/4 Hz roll rate. The gyro noise is included and the integration of the noise produces the random walking signal shown below.

$$\int \cos\phi d\phi = \sin\phi(t) = \int \frac{Q\omega_y^{\ b} + R\omega_z^{\ b}}{Q^2 + R^2} d\phi \qquad \text{EQ (C.1)}$$

$$\int \sin\phi d\phi = -\cos\phi(t) = \int \frac{R\omega_y^{\ b} - Q\omega_z^{\ b}}{Q^2 + R^2} d\phi \qquad \text{EQ (C.2)}$$

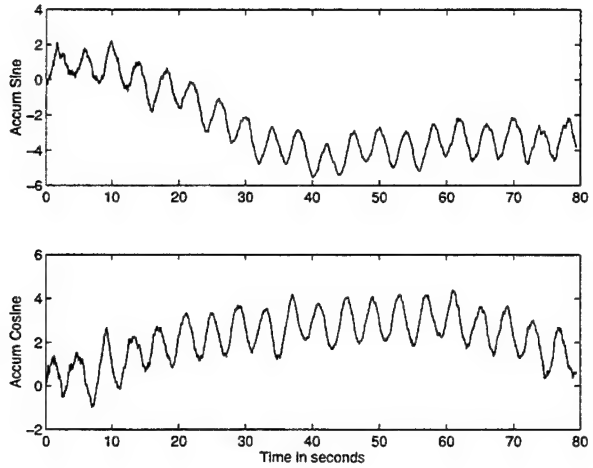


FIGURE C.1 Accumulated Sine and Cosine Roll from Algorithm

A different technique was implemented which involves estimating the maximum and minimum points for both EQ C.1 and EQ C.2 . This information provides t_{max} or t_{min} which pinpoints the roll angle when $\phi = 0$ and π for the cosine curve and when $\phi = \pi/2$ and $3\pi/2$ for the sine curve. Between the max and min values the roll angle is estimated using the roll rate estimate ω_x^b. EQ C.1 and EQ C.2 estimate roll from the maximums and minimums coming from the accumulated cosine curve in equation EQ C.1 . A similar set of equations is used to estimate roll from the accumulated sine curve of equating EQ C.2 .

$$\phi(t) = \frac{\pi}{2} + \omega_x^{\ b} \cdot (t - t_{max}) \qquad \text{EQ (C.1)}$$

$$\phi(t) = \frac{3\pi}{2} + \omega_x^{\ b} \cdot (t - t_{min}) \qquad \text{EQ (C.2)}$$

Data quantization rate plays a big roll in this algorithm. Finding the maximums and minimums required estimating the slope of the accumulated sine signal in Figure C.1 . The noise evident near the peaks required data smoothing by quantization and averaging. The error in the roll estimate is equal to the error in the estimated spin rate times the error in the estimation of t_{max} or t_{min}. Data quantization of 20 seconds at 100Hz (0.2 seconds Δt) yielded the t_{max} and t_{min} estimates shown in Figure C.2 .

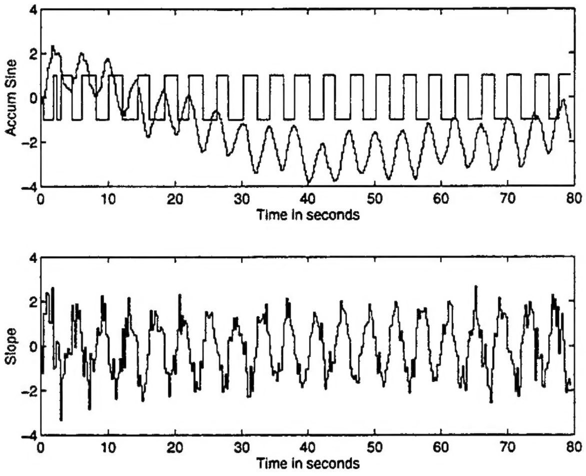


FIGURE C.2 T_{max} and T_{min} Estimates from Accumulated Sine

Since the data quantization Δt was 0.2 seconds and the roll rate was 1/4 Hz, the maximum error in from this algorithm should lie between + or - 18 degrees (not including errors in estimating the spin rate). Figure C.3 shows the roll error and confirms this is indeed the accuracy. Better accuracy could be achieved by increasing the data rate to 1000Hz and thus decreasing Δt to 0.02 seconds. This could potentially provide a factor of 10 improvement.

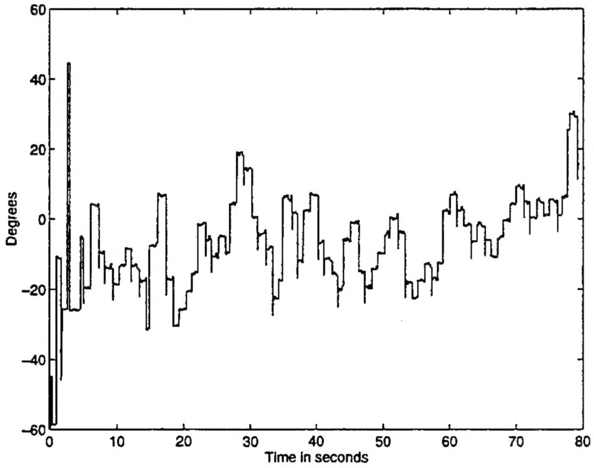


FIGURE C.3 Roll Error for Maximum/Minimum Algorithm (Data at 100Hz)

References

[1] Dowdle, J., "Competent Munitions Pre-ATD System Concept Description", Draper Laboratory Memo, Charles Stark Draper Laboratory, Cambridge, MA, Nov. 1994.

[2] Evans, A., Hermann, B., "Feasibility of Using a GPS Receiver to Navigate a Naval Gun-Launched Projectile to a Target", Technical Report, Naval Surface Warfare Center, Dahlgren, VA, 1993.

[3] Gustafson, D.E., information from a personal conversation, Charles Stark Draper Laboratory, Cambridge, MA, August 1993.

[4] Gustafson, D.E., "Preliminary Navigation Filter Sizing Study for ERGM", Draper Laboratory Memorandum E43-95-051, Charles Stark Draper Laboratory, Cambridge, MA, March 1995.

[5] Lieske, F., Reiter, M., "Equations of Motion for a Modified Point Mass Trajectory", U.S. Army Material Command, Ballistic Research Laboratories Report No. 1314, Aberdeen Proving Ground, MD, March 1966.

[6] Meriam, J., *Dynamics*, John Wiley and Sons, Inc., New York, NY, 1975.

[7] "Standard Gyro Terminology", Gyro and Accelerometer Panel, Aerospace and Electronic Systems Society, IEEE, 14 Jan 1974.

[8] Strang, Gilbert, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA 1986.

[9] Trott, M., "Course 6:432: Stochastic Processes", course notes, Massachusetts Institute of technology, Cambridge, MA, Spring 1993.

[10] Wetherbee, L., "Optimal Estimation of the Vehicle State in an Embedded Doppler/GPS Navigation System", MIT Masters of Science Thesis, Draper Laboratory Document CSDL-T-1223, Charles Stark Draper Laboratory, Cambridge, MA, May 1994.

[11] Wiles, G., "Tracking Projectiles: The GPS Artillery Registration Fuze Program", *GPS World*, September 1992.

[12] Willsky, A., "Course 6.433: Recursive Estimation", Supplementary Notes, Department of Electrical Engineering and Computer Science, Massachusetts Institute of technology, Cambridge, MA, Spring 1994.